Benutzerhandbuch Automatisierungs- und Fernwirksystem @250

Version 2.11

Lizenzbestimmungen

Die in diesem Handbuch beschriebene Software wird Ihnen gemäß den Bedingungen eines Lizenzabkommens zur Verfügung gestellt und darf nur unter den darin beschriebenen Bedingungen eingesetzt werden.

Daten, Abbildungen, Änderungen

Daten und Abbildungen sind unverbindlich. Änderungen, die dem technischen Fortschritt dienen, sind vorbehalten. Falls Sie Verbesserungs-– oder Änderungsvorschläge haben oder Fehler in dieser Druckschrift entdecken sollten, bitten wir um Ihre Mitteilung.

Gewährleistung

KEINE GEWÄHRLEISTUNG. Die technische Dokumentation wird Ihnen OHNE Mängelgewähr geliefert. OHP übernimmt keine Gewährleistung für deren Genauigkeit oder Verwendung. Die Verwendung der technischen Dokumentation oder der darin enthaltenen Informationen hat der Benutzer zu verantworten.

Schulung

Zur Vermittlung ergänzender Systemkenntnisse werden von OHP entsprechende Schulungen angeboten.

Warenzeichen

OHP, das OHP-Logo und ProWin sind Marken der OHP GmbH.

Die übrigen in diesem Handbuch verwendeten Produktnamen können eingetragene Warenzeichen und/oder Warenzeichen der jeweiligen Unternehmen sein.

Copyright

Kein Teil dieser Dokumentation darf ohne schriftliche Genehmigung der OHP GmbH in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Die Übersetzung in eine fremde Sprache ist nicht gestattet.

Copyright © 2007 OHP GmbH. Alle Rechte vorbehalten

Gedruckt in Deutschland.

Inhalt

Teil I Allgemeines 7

1 Gültigkeitshinweis 7

2 Einführung 7

2.1 Systemneutrale Automatisierungsplattform 7
2.2 Migration von AEG/Schneider Automatisierungs- und Fernwirktechnik 7

3 Systemvoraussetzungen 8

3.1 Systemvoraussetzungen Programmiersoftware 8
3.2 Systemvoraussetzungen Steuerung 9
3.2.1 @250 Systemkomponenten und E/A Vergleichsliste 9

4 Kommunikationsschnittstellen des @250 Systems 11

4.1 Kommunikationsprozeduren auf der ALU504/518: 11
4.2 Kommunikationsprozeduren mit KOS Koppelbaugruppen 11
4.3 Modembaugruppen für Fernwirkprozeduren 11
4.4 Ankopplung über Modbus 12
4.4.1 ModbusTCP Client 12
4.4.2 Modbus RS232 Slave / Modbus TCP-Server 12

Teil II Projektierung und Programmierung 13

1 Neue Leistungen der ALU 520 Firmware 14

2 Erzeugen eines neuen Projekts mit MULTIPROG 20

3 Ändern der IP-Adresse für die Verbindung zur ALU 520 21

4 Definition der Ein- und Ausgaben 22

4.1 Ein- und Ausgaben mit dem OHP SPS-Konfigurator 22
4.1.1 Der Dialog SPS-Konfigurator im Detail 24
4.2 Ein- und Ausgaben mit der Multiprog IO_Configuration 26
4.2.1 Definition Standard I/O Module 28
4.3 I/O Konfiguration für IEC 60870-5-101 oder IEC 60870-5-104 30
4.4 I/O Konfiguration für Modbus TCP Client 32
4.5 Projektierung des Shared-Memory-Bereichs für Modbus RS232-Slave / TCP-Server 34

5 Systemmerker 35

5.1 Systemmerker Projekt 35
5.2 Systemmerker Board 35
5.3 Systemmerker Service 36
5.4 Systemmerker Steckplatz in I/O Konfiguration belegt 37
5.5 Systemmerker Baugruppenausfall 37

5.6 Systemmerker Baugruppenstörung 38
5.7 Systemmerker Statusbyte 38
5.7.1 Definition Statusbyte KOS140 39
5.7.2 Uhrzeitübergabe durch KOS 140 39
5.7.3 Definition Statusbyte KOS141 40
5.7.4 Definition Statusbyte Modbus 41

6 Projektierung der Initialisierungsdateien 42

6.1 Einstellen der IP-Adresse auf der 504/518 42
6.1.1 Einstellen der IP-Adresse auf der ALU504/518 42
6.1.2 Zugriff über FTP oder TELNET 42
6.1.3 Zeitsynchronisation über GPS 43
6.2 Freigabe der Service-Routinen und Interfaces 44
6.2.1 Definition von Shared Memory 45
6.2.2 Modbus RS232 45
6.2.3 Modbus TCP-Server 46
6.2.4 Service-Routinen 46
6.3 IEC 60870-5-101 oder -104 Kommunikation 47
6.3.1 Allgemeines 47
6.3.2 Parameterdatei für die Kommunikationsprojektierung IEC 60870-5-104 (SVC_COM?) 48
6.3.3 Parameterdatei für die Kommunikationsprojektierung IEC 60870-5-101 (SVC_COM?) 49
6.3.3.1 Erläuterung zu den einzelnen Parametern im Standleitungs-Betrieb 50
6.3.3.2 Erläuterung zu den einzelnen Parametern im AWD-Betrieb 51
6.3.4 Parameterdatei für die Datenpunktdefinition bearbeiten (SVC_RTU?) 52
6.4 Projektierung der Initialisierungsdateien für Modbus RS232-Slave oder TCP-Server 60
6.4.1 Erläuterung zu den einzelnen Parametern 61

7 Firmwarebibliotheken 64

7.1 Allgemeines 64
7.2 Bibliothek ALUX20_OHP4_SYS 65
7.2.1 Auslesen und Stellen der ALU Systemzeit im Anwenderprogramm 65
7.2.2 Status der ALU Systemzeit lesen 66
7.2.3 Status der ALU Systemzeit auf ungültig setzen 67
7.2.4 Status der ALU Systemzeit bei GPS-Synchronisation 67
7.2.5 Status der ALU Systemzeit bei SNTP-Synchronisation 68
7.2.6 Systemtakte 68
7.2.7 Status der GPRS-Verbindung 69
7.2.8 SystemTick 70
7.2.9 Modbus-Server 70
7.3 Bibliothek ALUX20_OHP4_SYS_SA 71
7.3.1 Baugruppentyp auslesen bei DEA-Kopplung 71
7.3.2 Baugruppen-Statusmerker 72
7.3.3 Baugruppen-Statuswort 73
7.4 Bibliothek ALU352 OHP4 AWPWDG 75
7.4.1 Hardware-Watchdog über Anwenderprogramm 75

8 Tipps und Tricks 77

8.1 Update eines bestehenden Multiprog Projekts für Online-Ändern 77
8.2 Auslesen und Stellen der ALU504/518 Systemzeit im Anwenderprogramm 78
8.3 Uhrzeit stellen und ALU-Statusinformationen über Telnet 79
8.4 ALU-Statusinformationen über Telnet 80

TEIL III Baugruppenbeschreibungen 83

1 Baugruppenbeschreibung ALU 504/518/520 85

1.1 Schnittstellen 85
1.1.1 Schnittstellenbelegung der Schraubklemmen: 85
1.1.1.1 Stromversorgung 85
1.1.1.2 Schnittstelle für GPS Uhrzeitempfänger 86
1.1.1.3 CAN Schnittstelle (in Vorbereitung) 86
1.1.2 Serielle RS232 Schnittstellen: 86
1.1.3 Compact Flash Speicherkarten 86
1.1.4 ALU504/518 Anzeigeelemente 87
1.1.5 ALU504/518 Bedienelemente 88
1.2 Technische Daten ALU504/518 89

2 Baugruppenbeschreibung ALU 518-1/SFB und ALU 504-1/SFB 91

2.1 Allgemeines 91
2.2 Bedienung / Darstellung 92
2.3 Projektierung 92
2.3.1 Koppel-Betrieb 93
2.3.2 System-Feldbus-Anschluss 94

Teil I Allgemeines

1 Gültigkeitshinweis

Diese Beschreibung gilt für die ALU 5xx ab Firmwareversion 3.6 und MULTIPROG Plus Version 2.8.

Die aktuelle Firmwareversion stellen wir auf www.ohp.de im Kundenbereich als Download zur Verfügung.

Teilleistungen werden möglicherweise von älteren Firmwareversionen nicht unterstützt. Eine Aufstellung der Firmwareversionen und die zugehörigen neuen Leitungen finden Sie in **TEIL II Projektierung und Programmierung,** Kapitel 1.

2 Einführung

Das Automatisierungs- und Fernwirksystem @250 versteht sich als Weiterentwicklung der bewährten A250 Automatisierungs- und Fernwirktechnik und sichert die langfristige Verfügbarkeit der A250 Baureihe als leistungsfähige Automatisierungsplattform.

Das Automatisierungs- und Fernwirksystem @250 kann neben der Ausrüstung von Neuanlagen insbesondere auch zur Migration und damit zur Investitionssicherung von vorhandenen Anlagen eingesetzt werden, da sich das System nahtlos in die E/A Welt der A250 Baureihe einfügt.

2.1 Systemneutrale Automatisierungsplattform

Das @250 System ist als systemneutrale Automatisierungsplattform ausgelegt, die auf herstellerspezifische Lösungen soweit als möglich verzichtet. Basierend auf modernster offener embedded Industrie-PC Technik werden alle PC Standard-Schnittstellen wie z.B. Ethernet, RS232, USB, VGA, CF-Card unterstützt.

Mit der herstellerunabhängigen IEC 61131-3 Programmiersoftware MULTIPROG ist ein leistungsfähiges und ausgereiftes Programmiertool implementiert, welches bereits heute von über 70 namhaften SPS Firmen erfolgreich eingesetzt wird.

Mittels genormter Kommunikationsprotokolle wie der IEC 60870-5-101/104, oder dem weit verbreiteten ModbusTCP Protokoll und auch den optionalen OPC Server kann das System in verschiedenste Automatisierungs- bzw. Fernwirkumgebungen integriert werden.

Über die integrierte Ethernet- bzw. CAN-Schnittstelle können sowohl systemeigene E/A Erweiterungen über 1/SFB Koppler als auch handelsübliche E/A Systeme z.B. von Phoenix Contact über Ethernet TCP angeschlossen werden. Über diese E/A-Erweiterungen können so maximal bis zu 1.216 E/A Punkte verarbeitet werden.

2.2 Migration von AEG/Schneider Automatisierungs- und Fernwirktechnik

Die A250 Geräte bilden im oberen Anwendungssegment eine bewährte Automatisierungs- und Fernwirkplattform, in deren Installation erhebliche Investitions- und Zeitaufwendungen geflossen sind. Mit dem @250 System hat OHP eine Strategie entwickelt, die bei minimierten Kosten und Zeitaufwand die @/U-Geräte in eine neue und langfristig verfügbare Automatisierungs- und Fernwirkbaureihe migriert.

Bestechender Vorteil dieser Strategie ist die Kostenersparnis durch Nutzung vorhandener Peripherie wie E/A-Module und KOS-Baugruppen mit vorhandener Verdrahtung: Es wird nur ausgetauscht, was ausgetauscht werden muss. Dies ist in der Regel nur die Zentraleinheit, die anstelle einer ALU15x durch eine ALU5xx gewechselt wird.

Durch diese Maßnahmen stehen die neuen genormten IEC 60870-5-101/-104 Prozeduren mit moderner Programmiertechnik unter IEC 61131-3 zur Verfügung. Durch die Nutzung der vorhandenen KOS-Baugruppen ergebenen sich elegante Upgradestrategien von 1/F- und 1/W-Prozeduren auf genormte IEC-Prozeduren.

3 Systemvoraussetzungen

3.1 Systemvoraussetzungen Programmiersoftware

Hardware

Für die optimale Leistungsfähigkeit des SPS-Programmiersystems MULTIPROG müssen mindestens die folgenden Hardware-Voraussetzungen erfüllt sein:

Gerät/Baugruppe	Minimal	Empfohlen ab
IBM-kompatibler PC mit Pentium-Prozessor	Pentium II 350 MHz	Pentium III 500 MHz
Arbeitsspeicher	64 MB	128 MB
Festplatte	250 MB freier Speicherplatz	
CD-ROM-Laufwerk	benötigt	
VGA-Monitor Farbeinstellungen Auflösung	 256 Farben 800 x 600	 True Color 1024 x 768
Ethernet-Schnittstelle	benötigt	

Software

MULTIPROG benötigt eines der folgenden Betriebssysteme:

- Microsoft Windows 2000
- Microsoft Windows XP

3.2 Systemvoraussetzungen Steuerung

- Als Baugruppenträger kann sowohl das DTA 112 als auch das DTA 113 verwendet werden.
- Steckplatz 1 ist ausschließlich für die Zentraleinheit ALU 504/518 reserviert
- Die auf Steckplatz 2 befindliche Versorgungsbaugruppe DNP 106 wird nicht mehr benötigt. Dieser Steckplatz kann als zusätzlicher E/A-Steckplatz genutzt werden.
- Als Erweiterungsbaugruppenträger kann ein DTA 112 oder DTA 113 verwendet werden.
- Die Firmwareversion vorhandener DEA 106 sollte den Änderungsindex 20 haben (kann vom Beschriftungsschild des EPROMS auf der DEA abgelesen werden). Bei älteren Firmwareständen kann es zu Problemen in der E/A-Kopplung kommen. Wenn nach der Umrüstung Ihre Anlage dieses Problem auftritt, wenden Sie sich an die OHP-Kundenbetreuung um ein Update für die Firmware zu erhalten. EPROMS mit der Beschriftung "DEA105 Software Bitbus 042.217453.08" müssen auf jeden Fall ausgetauscht werden.
- Es werden die Standard E/A-Karten der A250 unterstützt (siehe Tabelle).
- Die OHP E/A-Baugruppen der 500er Serie können eingesetzt werden (siehe Tabelle).
- Über ModbusTCP Client können externe E/A-Komponenten und Systemerweiterungen z.B. von Phoenix Contact, ABB angeschlossen werden.

3.2.1 @250 Systemkomponenten und E/A Vergleichsliste

Nachfolgend sind die Systemkomponenten für das Automatisierungs- und Fernwirksystem @250 aufgeführt. Insbesondere auch die E/A Baugruppen, die klemmenkompatibel zu den Schneider Electric-E/A-Baugruppen sind.
Die aufgeführten OHP E/A-Baugruppen sind z. Zt. noch nicht vollständig verfügbar. OHP behält sich bei den E/A-Baugruppen vor, Vergleichstypen von Schneider Electric zu liefern. E/A-Baugruppen, die sich nicht in der Liste befinden, können ggf. auf Anfrage geliefert bzw. integriert werden.

Bezeichnung	Schneider Vergleichstyp	Beschreibung	
		Zentraleinheit, Grundbaugruppenträger	
ALU504 ALU518 ALU504-1SFB ALU518-1SFB		Zentraleinheit für @250 System	
DTA 512	DTA112	Baugruppenträger für Zentraleinheit und 4 E/A Baugruppen	
DTA 513	DTA113	Baugruppenträger für Zentraleinheit und 9 E/A Baugruppen	
MP@PLUS		MULTIPROG IEC 61131-3 Programmiersoftware für @120 System. AWL, FBS, KOP,ST und AS Querübersetzung AWL-KOP-FBS, Offline Simulation, Watch Fenster, Debugfunktionen, Projektvergleich, Querverweise.	

		Digitale Eingabe	**Eingangsspannung**
DEP512	DEP112	4 x 8 Digitale Eingänge, potentialgetrennt	24V DC
DEZ161		4 x 8 Digitale Eingänge, potentialgetrennt, intelligente Baugruppe für Echtzeitverarbeitung, Zählwerterfassung	24V… 60V DC
		Digitale Ausgabe	**Ausgangsspannung**
	DAP106	16 Relaisausgänge (Schiesser) mit Anzeige,	24…110V DC, 24…230V AC
DAP512	DAP112	4 x 8 Digitale Ausgänge, potentialgetrennt	24V DC, 0.5A
	DAP116	4 x 4 Digitale Ausgänge, potentialgetrennt	24V DC, 2A
		Digital kombinierte Baugruppen	**Ein/ Ausgangsspannung**
	DAP102	2 x 8 Digitale Eingänge, potentialgetrennt 2 x 8 Digitale Ausgänge, potentialgetrennt	24V DC 24V DC, 2A
DAP503	DAP103	2 x 8 Digitale Eingänge, potentialgetrennt 2 x 8 Relaisausgänge, potentialgetrennt	24…110V DC, 24…230V AC
	DAP104	8 Digitale Eingänge, potentialgetrennt 8 Relaisausgänge (Schliesser), potentialgetrennt, mit Anzeige	115/230V AC, umsteckbar 2.0A (4.0A kurzzeitig) 230V AC, 24V DC
		Analoge Eingabe	**Eingangssignal**
ADU515	ADU115	16 2polige oder 8 4polige, analoge Eingänge, 12Bit + VZ	Spannung +/- 10V, +/-5V, +/-1v, +/- 0.5V, +/- 50mV, Strom +/- 20mA, +/- 10mA, +/- 1 mA
ADU516	ADU116	16 analoge Eingänge, potentialgetrennt, 10Bit + VZ	+/- 10V, +/- 20mA
		Analoge Ausgabe	**Ausgangssignal**
	DAU104	4 4polige und 4 2polige oder 8 2polige analoge Eingänge, ohne Potentialtrennung, 12Bit + VZ 4 2polige analoge Ausgänge	PT 100 Geber, +/- 10V, +/- 5V, +/-20mA +/- 20mA, +/-10mA, +/-10V, +/-5V, +/-2V, +/- 1V, +/- 500mV
DAU509	DAU109	8 analoge Ausgänge, potentialgetrennt, 11Bit + VZ	+/- 10V, +/- 20mA,
		Koppelbaugruppen	**Schnittstelle**
KOS140	KOS140	Kommunikationsmodul für Fernwirkprotokoll Modnet 1/F.	1 x RS232
KOS141	KOS141	Kommunikationsmodul für Fernwirkprotokoll Modnet 1/W.	2 x RS232
KOS560		Kommunikationsmodul für Fernwirkprotokoll IEC 60870-5-101	2 x RS232
BIK514	BIK114	Kommunikationsbaugruppe für Modnet 1/SFB, E/A + Daten, max. 2 MBit/s	1 x RS485
DEA506	DEA106	Kommunikationsbaugruppe für Modnet 1/SFB, E/A , max. 2 MBit/s	1 x RS485
	DEA116	Kommunikationsbaugruppe für Modnet 1/SFB, E/A , max. 2 MBit/s	1 x RS485

4 Kommunikationsschnittstellen des @250 Systems

Das neue Automatisierungs- und Fernwirksystem @250 beinhaltet Kommunikationsschnittstellen, die entweder auf der ALU direkt oder über KOS-Koppelbaugruppen betrieben werden können.

4.1 Kommunikationsprozeduren auf der ALU504/518:

- IEC 60870-5-101
- IEC 60870-5-101 AWD für **A**utomatischen **W**ähl**d**ienst
- IEC 60870-5-104
- ModbusTCP Client / Server
- Modbus RS232 Slave

Die IEC Prozeduren entsprechen der Kompatibilitätsliste @250 für IEC 60870-5-101 und IEC 60870-5-104.

4.2 Kommunikationsprozeduren mit KOS Koppelbaugruppen

- Modnet 1/F (SEAB-1F) - KOS 540
- Modnet 1/W - KOS 541
- IEC 60870-5-101 - KOS 560
- IEC 60870-5-101 AWD - KOS 560

☞ **Hinweis:** Die KOS 140 bzw. KOS 141 bestehender Stationen wird unterstützt.

4.3 Modembaugruppen für Fernwirkprozeduren

Für das @250 System sind über die Kommunikationsprozeduren IEC 60870-5-101 – AWD und IEC 60870-5-104 folgende Modemankopplungen möglich:

- UEM001 - FSK Standleitungsmodem, 600/1200 Baud, CCITT-Raster.
  Einbauplatine für KOS 140/540 oder KOS 141/541
- UEM202 - FSK Standleitungsmodem, wie UEM 001, jedoch Hutschienenmontage
- UEM300 / UEM301 - Analog-Modem für Wählleitungsbetrieb
- UEM302 - ISDN-Terminal für Wählleitungsbetrieb
- UEM303 / UEM 304 - Dualband GSM Modem
- UEM306 / UEM 307 - GPRS Modem

4.4 Ankopplung über Modbus

4.4.1 ModbusTCP Client

Die ALU 504/518 ist mit einer ModbusTCP Client-Schnittstelle ausgerüstet, über die externe E/A Module oder Steuerungen (z.B. ABB 07 KT 97, Phoenix Contact Ethernet Inline Buskoppler FL IL 24 BK- oder FL IL 24 BK-B-PAC) die über eine ModbusTCP Server-Schnittstelle verfügen, angekoppelt werden können.

Anmerkung: Der Client entspricht in seiner Funktionalität dem Master bei einer seriellen Modbus-Ankopplung.

Weiterführende Informationen siehe **Projektierung und Programmierung** Kapitel 4.4.

4.4.2 Modbus RS232 Slave / Modbus TCP-Server

Die zwei seriellen Schnittstellen der ALU 504/518 können als Modbus Slave Schnittstelle projektiert werden. Der Modbus TCP-Server wird über die Ethernet-Schnittstelle angekoppelt.

Über diese Schnittstellen können beliebige Modbus-Master / Modbus-Clients Daten aus der Station abfragen. Bis auf die Auswahl des Interfaces (SERIELL oder TCP_IP) ist die Projektierung für beide Protokolle identisch.

Von der ALU 504/518 werden folgende Funktionscodes unterstützt:

FC 1	Lese Bit	Read Coils
FC 2	Lese Eingang	Read Discrete Inputs
FC 3	Lese Register	Read Holding Register
FC 4	Lese Eingangswort	Read Input Register
FC 5	Schreibe einzelnes Bit	Write Single Coil
FC 6	Schreibe einzelnes Register	Write Single Register
FC 15	Schreibe Bitbereich	Write Multiple Coils
FC 16	Schreibe Registerbereich	Write Multiple Register

Die Daten werden im Shared Memory Bereich zur Verfügung gestellt. Die Projektierung erfolgt über INI-Dateien, die auf die CF-Karte gespeichert werden.

Anmerkung: Die Funktionen FC 2 und FC 4 greifen nicht direkt auf die Eingänge der Hardware zu. Die Eingänge müssen auch hier auf dem Shared Memory Bereich abgebildet werden.

Anmerkung: Der Modbus TCP-Server unterstützt maximal 24 Client-Verbindungen. Hierbei ist zu beachten, dass der Modbus-Client einer @120 oder @250 jeweils eine Verbindung für das Lesen und eine für das Schreiben von Daten herstellt. Die Anzahl der Transaktionen (gleichzeitig gesendete Abfragen bei asynchronem Betrieb) ist auf 4 begrenzt.

Weiterführende Informationen siehe **Projektierung und Programmierung** Kapitel 4.5, Kapitel 6.2 und Kapitel 6.4.

Teil II Projektierung und Programmierung

Hinweis: Wenn im folgenden Text der Begriff ALU520 verwendet wird, sind damit alle ALU-Typen der @250 Serie gemeint.

Die Projektierung und Programmierung wird anhand eines Projekts erläutert, das mit der Vorlage "ALU520 " erstellt wurde.

Diese Vorlage kann von der Internetseite der Firma OHP Automation Systems GmbH geladen werden.

Die Vorlage beinhaltet folgend Beispiele:

- die I/O Konfiguration für eine Eingabebaugruppe DEP 112 und eine Ausgabebaugruppe DAP 112
- einen Funktionsbaustein für die Bildung von umlaufenden 16 Bit Zählwerten für digitale Eingänge
- ein Programm "Main" als Funktionsplan, in dem die Baugruppen-Ausfallmerker für die DEP 112 und DAP 112 zu einem Sammelfehler zusammengefasst werden.
- die globalen Variablen für Statusinformationen der PAB-Teilnehmer im Zentralbaugruppenträger

1 Neue Leistungen der ALU 520 Firmware

Die aktuelle Firmware-Version kann über das Multiprog Menü "Online – Projektkontrolle – Info" ermittelt werden.

Hinweis: Da die Treiber für ALU 520 und ALU 320 zum Teil aus gleichen Quellprogrammen gebunden werden, kann sich der Versionsindex erhöhen, auch wenn die jeweiligen ALU-Treiber von dieser Änderung nicht betroffen sind. Aus diesem Grund kann es zu Versionssprüngen kommen.

Version 1.40

Es wurde eine zusätzliche Funktionalität bei den Treibern IEC 60870-5-101 und ...104 eingeführt. Für Sollwerte und Bitstrings in Befehlsrichtung kann jetzt ein Initialwert vorgegeben werden. Dieser Initialwert kann auf zwei Arten verarbeitet werden:

Modus 1:
Der projektierte Initialwert wird einmalig bei Kaltstart in die entsprechenden %IW bzw. %ID Variablen eingetragen.
Sobald ein Telegramm vom Leitsystem eintrifft, wird der Wert des Telegramms solange in die entsprechende Variable eingetragen, bis ein neuer Wert vom Leitsystem eintrifft.

Modus 2:
Der projektierte Initialwert wird bei Kaltstart in die entsprechenden %IW bzw. %ID Variablen eingetragen.
Trifft ein Telegramm vom Leitsystem ein, wird der Wert des Telegramms in die entsprechende Variable eingetragen und anschließend wieder durch den projektierten Defaultwert überschrieben. Je nach Länge des Anwenderprogramms kann der Telegrammwert für 1 ... 5 Programmzyklen anstehen, bevor er durch den Initialwert überschrieben wird.
Auf diese Weise kann vom Anwenderprogramm erkannt werden, ob ein neues Telegramm vom Leitsystem geschickt wurde, auch wenn sich der Wert im Telegramm selbst nicht geändert hat.

Zwischen Modus 1 und 2 kann ebenfalls projektierbar gewählt werden.

Projektierungsbeispiel:

```
//: Bitstring C_BO_NA
1:VAR_CBO_NA_INIT= 0x0000FFFF
1:VAR_CBO_NA_CMD= TRUE     // TRUE = Modus 2, Initialwert nach jeder Telegrammübergabe

//: Sollwert C_SE_NA
1:VAR_CSE_NA_INIT= 0xFFFF
1:VAR_CSE_NA_CMD= FALSE     // FALSE = Modus 1, Initialwert nur bei Kaltstart
```

Version 1.54

Bitbus (BIK514/114) und DEA Ankopplung realisiert

Baugruppen DAP102, DAP103, ADU115, ADU116, DAU104, DAU108, DAI109, KOS140 in die Treiberdatei aufgenommen

Uhrzeitsynchronisation über GPS, IEC...101/104 Zeittelegramme und Anwenderprogramm.

Neue Parameter für SVC_RTUn.INI

RTU_TIME_SET: 0, 1 oder 2

0 = Zeit darf über IEC gestellt werden (default)
1 = Zeit darf nie über IEC gestellt werden

2 = Zeit darf nur über IEC gestellt werden, wenn aktuelle Zeit IV (ungültig en: invalid) ist

Der folgende Parameter ist nur gültig, wenn RTU_TIME_SET ungleich NULL ist!

RTU_TIME_ACK_NEG: FALSE oder TRUE
FALSE = IEC Zeittelegramm wird mit positiver Response bestätigt (default)
TRUE = IEC Zeittelegramm wird mit negativer Response bestätigt

Der folgende Parameter ist nur gültig, wenn RTU_TIME_SET = 2 ist!

RTU_TIME_SYNC_ENABLE: Eingabe in Minuten (1…65535)
Delta-Zeitangabe in Minuten, seit dem letzten Stellen der Uhr. Nach Ablauf darf die Uhr über IEC-Telegramm gestellt werden.

Version 1.61
Wenn ein ADU projektiert aber nicht gesteckt ist, wird der Messwerte jetzt mit 0 angezeigt (früher 32767).

V**ersion 1.63**
Treiber Zugriff für mehrer DEA-Magazin an einem Bus optimiert

Version 1.70

Die Namen der INI-Dateien wurden geändert, damit diese mit der neuen Bedienoberfläche über Multiprog erzeugt werden können.

Alter Name	**Neuer Name ab Version 1.70**
PCOS_A520.ini	PCOSA520.ini
Slot_Device.ini	SlotDev.ini
SVC_COMn.ini *)	SVCCOMn.ini *)
SVC_RTUn.ini *)	SVCRTUn.ini *)
Modbus_Server.ini	MBSRV.ini

*) n steht für die jeweils zugehörige Service-Routine (1…4)

Achtung: Wenn Sie eine neue Firmware ab Version 1.70 verwenden und Ihre INI-Dateien von älteren Firmwareversionen benutzen möchten, müssen Sie diese vor dem Speichern auf der CF-Karte umbenennen. Inhaltlich sind die Dateien gleich geblieben.

Version 1.74

Es können jetzt beide COM-Schnittstellen gleichzeitig für die IEC 60870-5-101 Kommunikation verwendet werden.

Es können jetzt zwei Service-Routinen für die Kommunikation mit IEC 60870-5-104 definiert werden.

Weiterführende Informationen siehe **Teil II Projektierung und Programmierung,** Kapitel 6.3.2

Version 1.75

KOS 141 und KOS 560 in Treiberdatei aufgenommen.

Version 1.76
Treiber für zweite Ethernet-Schnittstelle eingebunden.

Version 1.80
Firmwarebaustein für RS232 (SCOM) Kopplung eingebracht.
Bei Blocktelegrammen IEC...101/104 werden jetzt nur noch max. 127 Objekte vom Typ Einzel- oder Doppelmeldung geblockt.

Grenzen für Übergabeschnittstelle erweitert

IEC101
Maximal 2096 Objekte in Melde- und Befehlsrichtung in der SVCRTUn.INI Datei.
Maximal 512 Byte in Befehlsrichtung und 1024 Byte in Melderichtung für die I/O-Konfiguration

IEC104
Maximal 6000 Objekte für Service 1 und 2 und max. 2096 Objekte für alle weiteren Service-Routinen in der SVCRTUn.INI Datei.
Maximal 512 Byte in Befehlsrichtung und 1024 Byte in Melderichtung für die I/O-Konfiguration

Version 1.82
Anbindung BIK014 (ALU BIK) implementiert

Version 1.86
PAB520.DLL für KOS140 wegen UZ251-Slave überarbeitet.

Version 1.87

- IEC101 AWD Betrieb: Routine für das Senden des Passworts geändert. Die Zeichen werden jetzt schneller nacheinander gesendet.
- Modbus TCP für mehrere Request freigegeben
- System Shared Memory auf max. 32KByte erweitert!
- Modbus Shared Memory auf max. 32KByte erweitert!
- PAB520.dLL - BIK114 Treiber für SEAB erweitert
- SSDKOSBIKA.dll und CDLSSDKOSBIKA.dll für SEAB-FWB erweitert
- SEAB Firmwarebibliothek für Multiprog bzw. ProConOS eingebunden

Version 1.89

- SSDKOSBIKA.dll und CDLSSDKOSBIKA.dll für SEAB-FWB erweitert
- IEC101/104: Ringpufferdaten werden bei Unterspannung auf CF-Karte geschrieben und bei Warmstart wieder geladen
- IEC101/104: Behandlung des OV-Bit bei Messwerten überarbeitet
- IEC101: Objekt-Nr. jetzt auf 3 Oktett erweitert

Version 1.90

- IEC101/104 ADU Messwertverarbeitung auf 2er Komplement umgestellt und Qual.- OV anstoss Übertragung geändert
- IEC101 AWD Betrieb:
  - AT_INIT String auf 28 Zeichen erweitert
  - 2 weitere AT_INIT Strings für SVCCOMn.ini zugelassen (AT_INIT2 und AT_INIT3)
- SEAB-FWB - Start/STOP Problem nach AWP Download beseitigt
- Identcode DAP106 hinzugefügt
- DEA Ablauf bei Fehlersynchronisation verbessert

Version 1.91

- ALUBIK Zugriffe (ACF Bit Faktor 32) modifiziert!
- A520/A320 Modbus Server Tcp/IP Kommunikation für mehrere Transaktionen optimiert.
- IEC101 AWD LED Ansteuerung eingebunden:
  Die Funktion ist nur für ein internes MODEM (COM3) vorgesehen.
  Einstellung in SVCCOM?.INI unter Section AWD101-MODE
  MODEM_LED=TRUE/FALSE

  Auswirkung bei TRUE:
  LED ON - Datenverbindung ("CARRIER") vorhanden - Passwort (Rx/Tx) und IEC-Kommunikation.
  LED OFF - keine Verbindung
  LED Blinken(Slow) - Verbindungsaufbau UST->LR
  LED Blinken(Fast) - Verbindungsaufbau LR (Ring)->UST
  LED 2 mal blinken (500ms) und Pause ... Modem Initialisierungsproblem
  LED 3 mal blinken (300ms) und Pause ... Modem nicht mehr bereit
- Bei IEC101-AWD das Senden des Passworts mit "CR" abschließen (default: TRUE)
  Deaktivieren dieser Funktionalität unter Section AWD101-MODE mit
  DIAL_PW_CR=FALSE

Version 1.92

Bei DEA-Ankopplung wurde der ADU-Vorzeichenfehler behoben.
ADU115/116 setzten bei positiven Werten das MSB (Bit15). Im Grundmagazin war dieses Problem bereits schon behoben. Gleiches gilt auch für die DAU109.

Version 1.94

Bei Telnet Verbindung - Willkommenstext mit Versionskennung implementiert

Version 3.00 (BETA Projektversion)

- Umstellung auf PROCONOS4 auf Basis der Version 1.94.
  (Voraussetzung für Online-Ändern)

Version 3.10 (BETA Projektversion)

- BIK314 eingebunden

Version 3.12 (BETA Projektversion)

- Auswertung der ALUX20.ini für SNTP Parameter "Server2" ergänzt.
  Wichtig: Der Parameter "SERVER2" muss immer hinter "SERVER" angegeben werden!
- Datum/Uhrzeit Überwachung für SNTP Dienst erweitert
- Diag_TSync.exe
  Diagnose für Datum/Uhrzeit Dienst SNTP Client (NTP0) etc. SNTP - 2 Serveradresse freigegeben.
- SNTP Service (Client)
  Zweite Serveradresse ergänzt. Die Serveradresse auf IP Adressebasis (xxx.xxx.xxx.xxx) erweitert, zuvor war nur Hostname möglich.
- UDPCFG Server
  Aktualisierung der TCP/IP Adresse. Uhrzeit stellen mittels Broadcast erlaubt!
- RAS Initialisierung - ALURAS.ini
  Initialisierung Sektion *AXRASVPN-PARAMS-T2* aktiviert

Version 3.16

- Offizielle Freigabeversion der neuen Leistungen ab 3.0

Version 3.18

- Abschaltverhalten für dezentrale E/A überarbeitet.
- PAB-DEA Ankopplung über Bitbus verbessert.
- ADU114 Ausgabefehler bereinig.

Version 3.19
Modbus TCP-Client

- Priorität der Modbus Threads auf PCOS Systemtask Niveau (121) angepasst.
- Verbindungsaufbau zu Modbus Servern optimiert (REACT).
- Überwachung Projektierungsparameter eingeführt.
- Überwachung der Response-Timeout Zeiten mit 10ms Faktor beaufschlagt.
- Anzahl max. Modbus Server auf 32 IP Adressen begrenzt.
- Anzahl I/O Gruppen auf max. 100 überwacht.
- Ablauf im Successive Mode (Response TOUT) überarbeitet.

Version 3.20

- Systemmerker Baugruppenstörung und Slot-Status (DWORD) für BIKX14 Modnet 1/SFB Status integriert.
- BIKx14 Ausfallmerker für reine Modnet 1/SFB Kommunikation überarbeitet.

Version 3.21

- Modbus TCP/IP Server: Anzahl Verbindungen von 24 auf 44 erhöht.

Hinweis: KW-Client benötigt 2 Verbindungen, d.h. eine für Read-Requests und eine für Write-Requests.

Version 3.40

- SystemTick geändert

  Parameter "KSYSTICKDIS= (TRUE/FALSE)" für PCOSAX20.ini in Sektion PCOS-SYSTEM aktiviert (def. FALSE) .
  KSYSTICKDIS = TRUE bewirkt, dass der Systemtick sich wie in den vorhergehenden Versionen verhält, d.h. zyklische Tasks brauchen die doppelte Intervallzeit!
  KSYSTICKDIS = FALSE bewirkt, dass der Systemtick halbiert wird, dies bewirkt, dass zyklische Tasks genau mit der projektierten Intervallzeit aufgerufen werden.
  Ist der Parameter aktiv, so erscheint in der Multiprog Versionsanzeige hinter V3.40 ein "S" (V3.40S) für ProConOS ohne Kernel SystemTick.

- PAB Treiber für KOS140/KOS141 bezüglich der zyklischen Befehlsübernahme überarbeitet

Version 3.41

- Speicherbereich für Shared Memory erweitert und für ALU-Typ skaliert

  ALU500: Shared Memory 8 kByte
  ALU504: Shared Memory 32 kByte
  ALU518: Shared Memory 64 kByte
  ALU520: Shared Memory 16 kByte

  Der Offsetparameter für den Shared Memory muss immer geradzahlig sein und kann zwischen 0 und 32768 definiert werden.
  Die Grösse (Anzahl Byte) des Shared Memory Bereichs muss immer geradzahlig sein und kann zwischen 0 und ALU-Typ Obergrenze definiert werden.

Version 3.43

- Firmwarebibliothek ALUX20_SBP_SEAB1N für SEAB1N Kommunikation eingeführt (zur Zeit nur Slave Funktionalität).

Version 3.44

- Systemmerker Baugruppenstörung und Slot-Status (DWORD) für BIKX14 Modnet 1/SFB Status integriert.
- BIKx14 Ausfallmerker für reine Modnet 1/SFB Kommunikation überarbeitet.

Version 3.45

- IEC 60870-5-103 eingeführt
  Modul mit IEC101 Datentypen im privaten Bereich der IEC103 realisiert.

Version 3.50

- Neue ProConOS Version (4.0.494) von KW-Software eingebunden. Verbesserung beim Online-Tauschen.

Version 3.60

- Hardware-Watchdog über Anwenderprogramm

2 Erzeugen eines neuen Projekts mit MULTIPROG

Beim Einrichten eines neuen Projekts gehen Sie folgendermaßen vor:

Schritt 1 Im Menü Datei "Neues Projekt…" anwählen.

Schritt 2 Vorlage "ALU520" mit Doppelklick auswählen.

Schritt 3 Gegebenenfalls die IP-Adresse für die Verbindung zur ALU 520 ändern.

Schritt 4 Projekt über Menü Datei "Projekt speichern unter…" mit Projektnamen abspeichern.

Schritt 5 Das neue Projekt kompilieren (F9 oder Menü Code -> Make). Dadurch werden die Datentypen und Funktionsbausteine in den jeweiligen Auswahlfenstern zur Verfügung gestellt.

Schritt 6 Definition der Ein- und Ausgaben über den SPS-Konfigurator vornehmen.

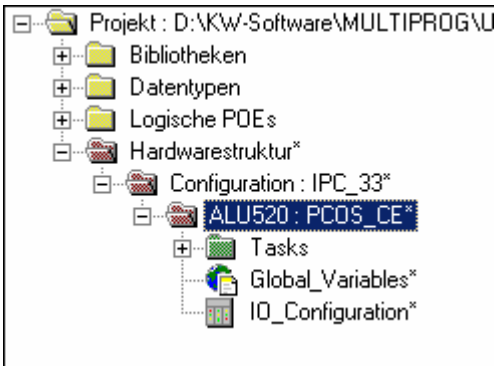
Schritt 7 Logischer Programm-Organisationseinheiten (POE) erstellen.

Schritt 8 Die erstellten POEs in die Liste der auszuführenden Tasks eintragen.

Schritt 9 Programmcode erzeugen.

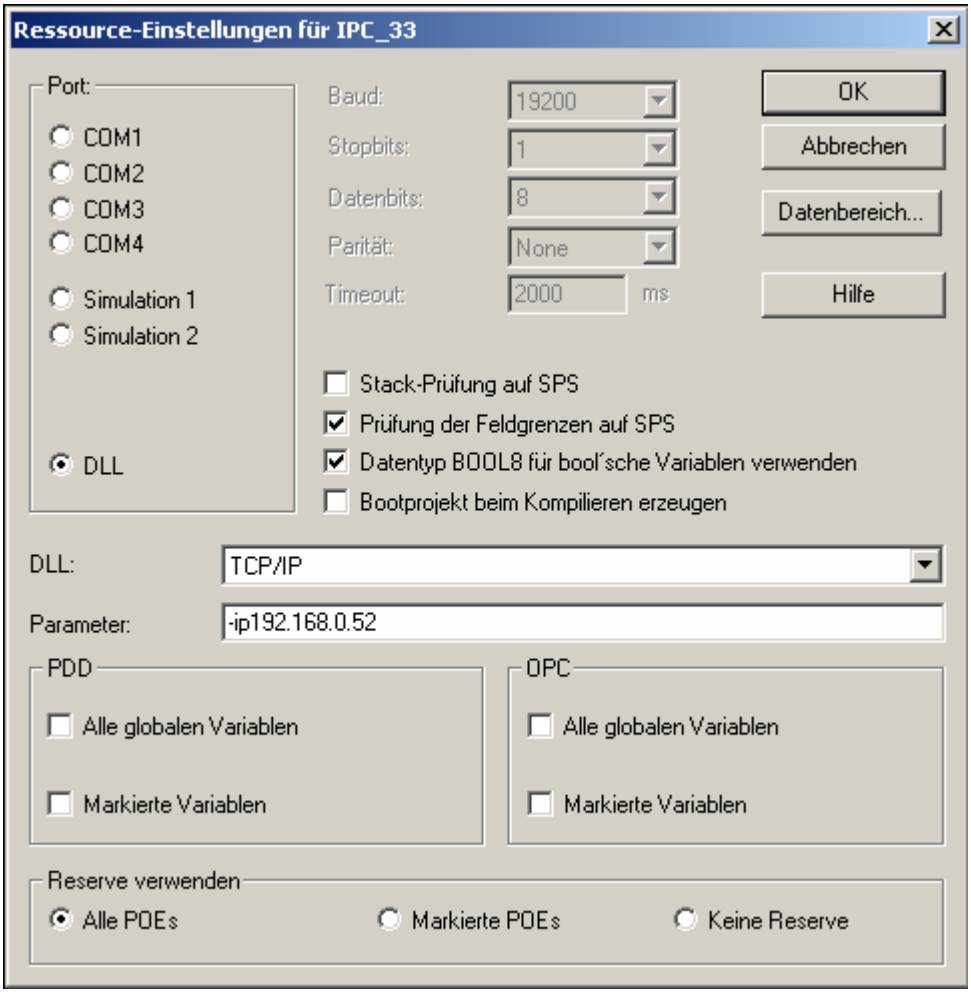
Schritt 10 Programmcode über Online-Funktion in die ALU 520 laden und starten.

3 Ändern der IP-Adresse für die Verbindung zur ALU 520



Wählen Sie in der Hardwarekonfiguration die Ressource, im Beispiel ALU520:PCOS_CE, mit der rechten Maustaste an. Öffnen Sie über das Pop-UP-Fenster den Dialog "Einstellungen… ".

Ändern Sie in der Zeile Parameter die IP-Adress ab.



4 Definition der Ein- und Ausgaben

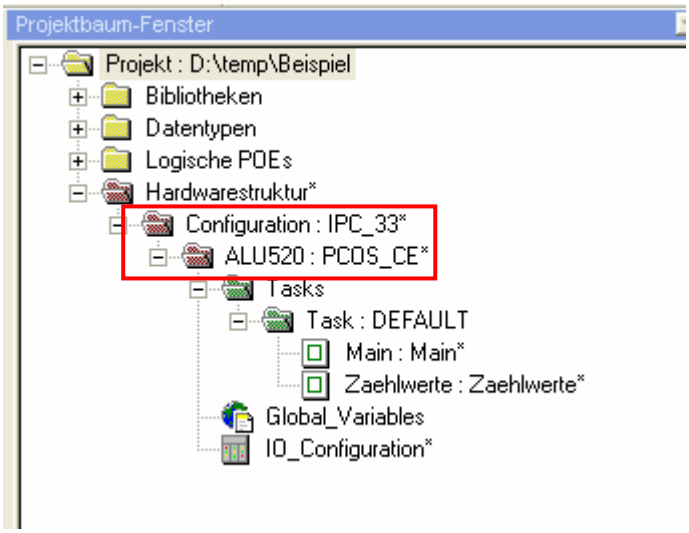
4.1 Ein- und Ausgaben mit dem OHP SPS-Konfigurator

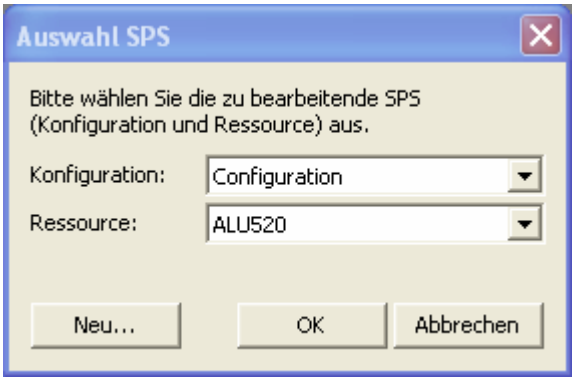
Sofern Sie Multiprog von der OHP Installations-CD inklusiver der OHP Add-Ons installiert haben, steht Ihnen der neue SPS-Konfigurator zur Verfügung.

Sollte dies nicht der Fall sein, können Sie diesen als registrierte Kunde von www.ohp.de aus dem Bereich Kundenservice – Update – Fernwirktechnik – Projektiersoftware laden.

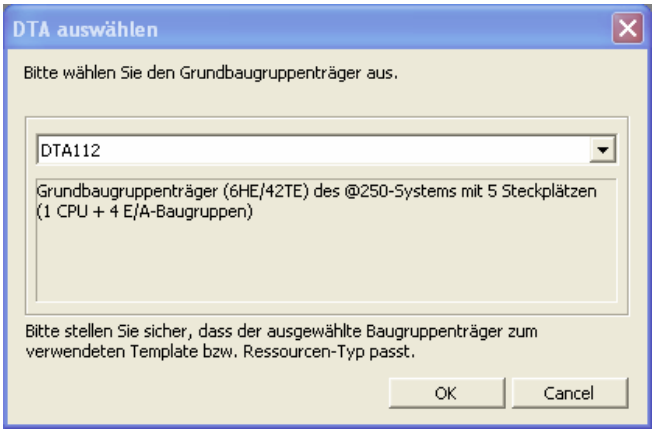
Mit Hilfe des neuen SPS-Konfigurators kann die E/A-Konfiguration einer Anlage vom Anwender schnell und komfortabel durchgeführt werden.

Ein Klick auf den Anwahlbutton "SPS-Konfigurator" in der Menüleiste öffnet den Dialog zur Auswahl der zu bearbeitenden SPS. Da eine SPS im Multiprog-Projektbaum durch einen Konfigurationsknoten und einen untergeordneten Ressourcen-Knoten repräsentiert wird, erfolgt die Auswahl über die entsprechenden Knotenbezeichnungen.

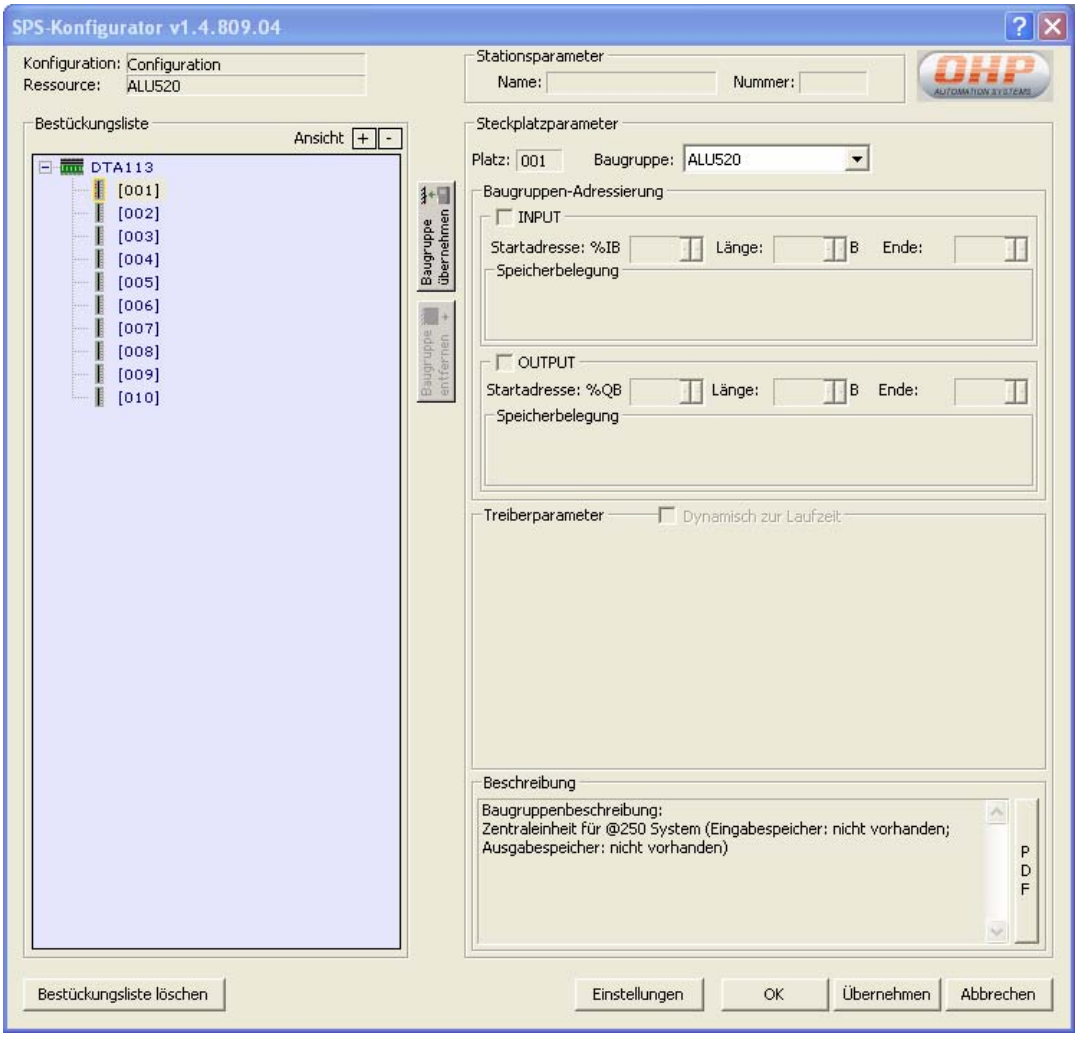




Nach Auswahl der zu bearbeitenden Konfiguration und Ressource und Klick auf Button "OK" wird eine bestehende Bestückungsliste geöffnet. Falls noch keine Bestückungsliste existiert, dann wird der Dialog "Auswahl Grundbaugruppenträger" aufgerufen:


DTA auswählen
Bitte wählen Sie den Grundbaugruppenträger aus.
DTA112
Grundbaugruppenträger (6HE/42TE) des @250-Systems mit 5 Steckplätzen
(1 CPU + 4 E/A-Baugruppen)
Bitte stellen Sie sicher, dass der ausgewählte Baugruppenträger zum
verwendeten Template bzw. Ressourcen-Typ passt.
OK
Cancel

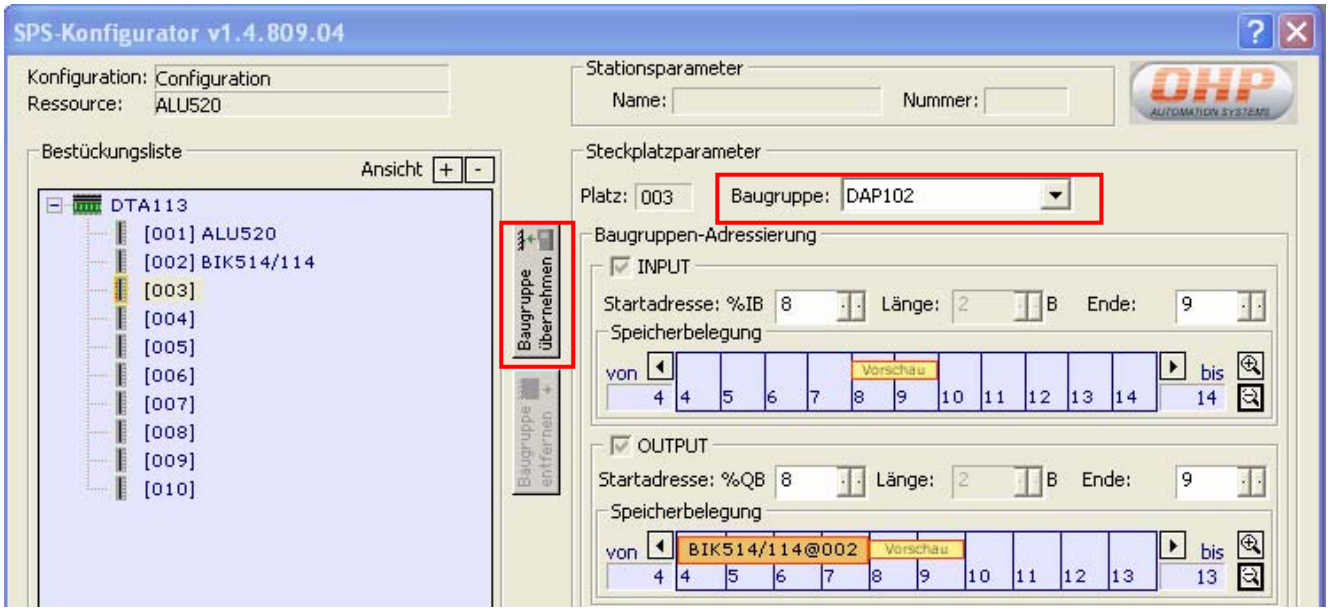
Dort wird für die Ressource ALU520 ein DTA112 oder DTA113 zur Auswahl angeboten und nach Klick auf Button "OK" erscheint der Dialog SPS Konfigurator:



4.1.1 Der Dialog SPS-Konfigurator im Detail

Wählen Sie jetzt für die einzelnen Steckplätze die entsprechenden Baugruppen aus der Listbox aus und übernehmen Sie diese mit dem Button "Baugruppe übernehmen" in die Bestückungsliste.

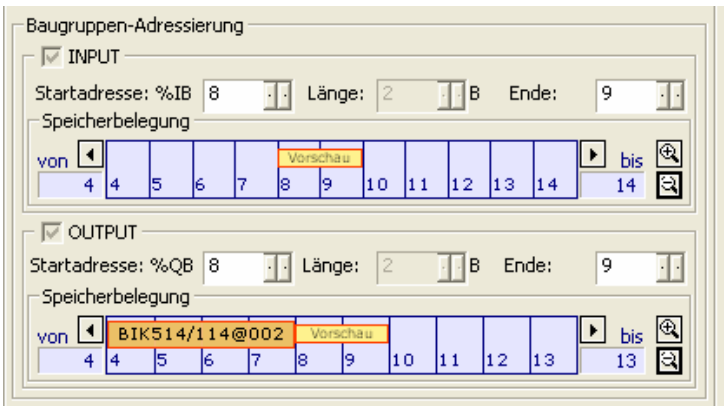
Auf dem ersten Steckplatz 001 kann im DTA112/113 nur eine ALU5xx gesteckt werden.



Der Speicherbereich jeder Baugruppe muss in dem Feld "Startadresse" festgelegt werden. Dies wird am Beispiel einer DAP 102 deutlich gemacht, die sowohl digitale Eingänge wie auch Ausgänge hat.

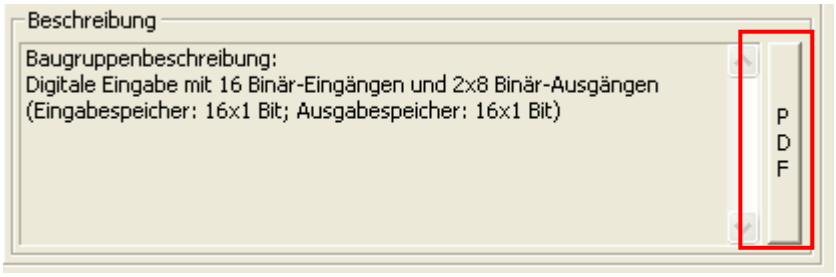
Durch das Eintragen einer 8 starten im folgenden Beispiel die benötigten 2 Bytes (2 * 8 Bits) der Eingänge ab Eingabe-Byte 8 (%IB8) und gehen bis Eingabe-Byte 9 (%IB9). Das gleiche gilt für die Ausgänge. Der SPS-Konfigurator schlägt automatisch den nächsten möglichen Ein- bzw. Ausgabebereich vor.

Der belegte Speicherbereich wird auch im Anzeigefeld "Speicherbelegung" als gelber Balken über den benutzten Byte-Adressen markiert.



Hinweis: Eine nachträgliche Veränderung z.B. der Startadresse, muss immer mit dem Button "Baugruppe übernehmen" bestätigt werden, damit die Einstellung übernommen wird.

Im Beschreibungsfeld wird eine Kurzanleitung zur jeweils angewählten Baugruppe angezeigt. Mit einem Klick auf den Button "PDF" kann eine detaillierte Beschreibung der OHP Baugruppe als PDF- Datei geladen werden, dazu muss das Programm "Adobe Reader" (mindestens in der Version 5.0) auf dem Programmiergerät installiert sein.

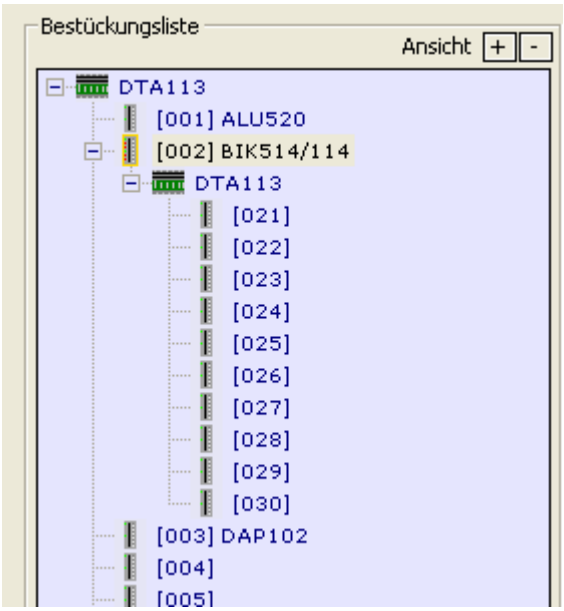


Um einen Erweiterungsbaugruppenträger zu projektieren, müssen Sie eine BIK 514/114 in die Bestückungslist eintragen.

Wählen Sie die BIK in der Bestückungsliste mit der rechten Maustaste an und klicken Sie im folgenden Dialog auf "Dezentralen Baugruppenträger (DTA) zuweisen…".

Anschließend wählen Sie das gewünschte DTA aus.

Auf diese Weise können einer BIK auch mehrere Erweiterungsbaugruppenträger zugewiesen werden.

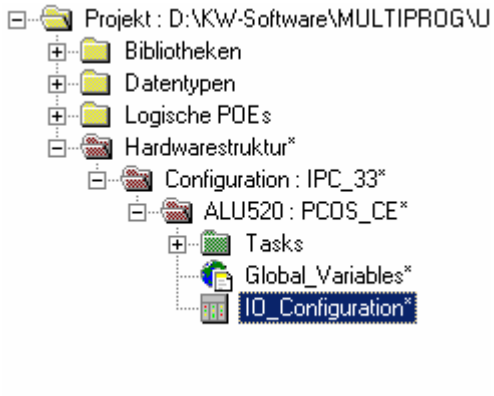


Im Erweiterungsbaugruppenträger können Sie wie zuvor beschrieben Ein- und Ausgabebaugruppen zuweisen.

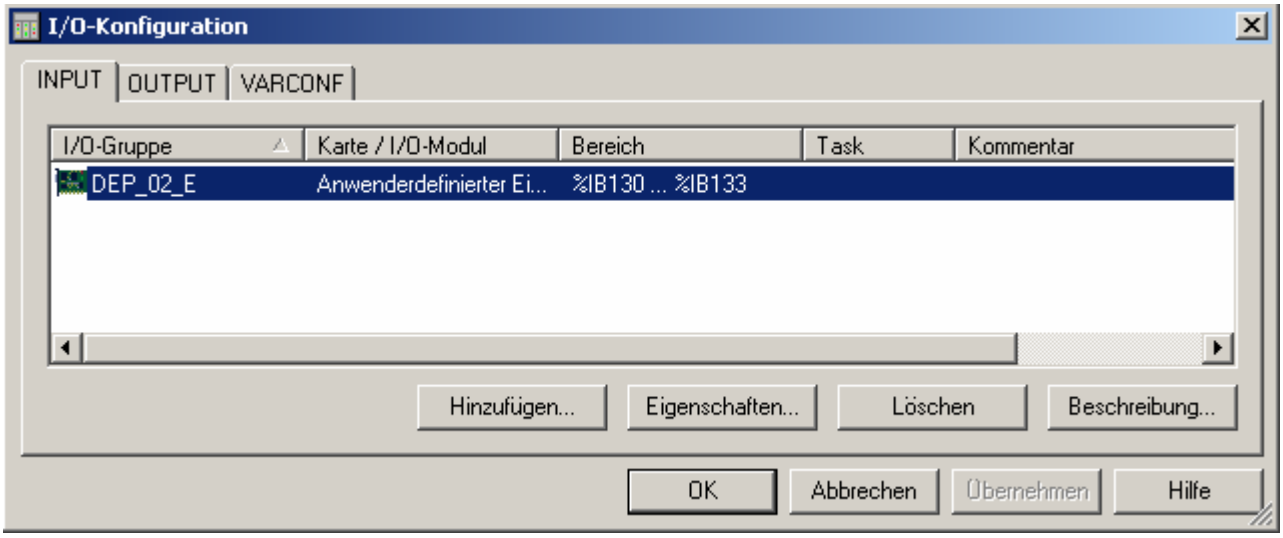
4.2 Ein- und Ausgaben mit der Multiprog IO_Configuration

Über die I/O-Konfiguration (Menü IO_Configuration) werden die Schnittstellen zu den Ein- und Ausgabebaugruppen, den Expertenbaugruppen und der ModbusTCP Kopplung definiert.

☞ **Hinweis:** Bei Einsatz von Erweiterungsbaugruppenträgern müssen Sie den SPS-Konfigurator für die Projektierung benutzen.



Die IO_Configuration ist ein Untermenü in der Hardwarestruktur und kann durch Doppelklick oder über das Pop-Up-Menü (Anwahl über die rechte Maustaste) geöffnet werden.



Achtung:

Der Name muss für jede Baugruppe eindeutig angegeben werden. Dies gilt auch bei gemischten Baugruppen wie der DAP 102 oder der KOS xxx. Hier muss jeweils ein Eintrag im Register INPUT und OUTPUT erfolgen, die Namen dürfen nicht identisch sein. Es können maximal 8 Zeichen eingegeben werden.

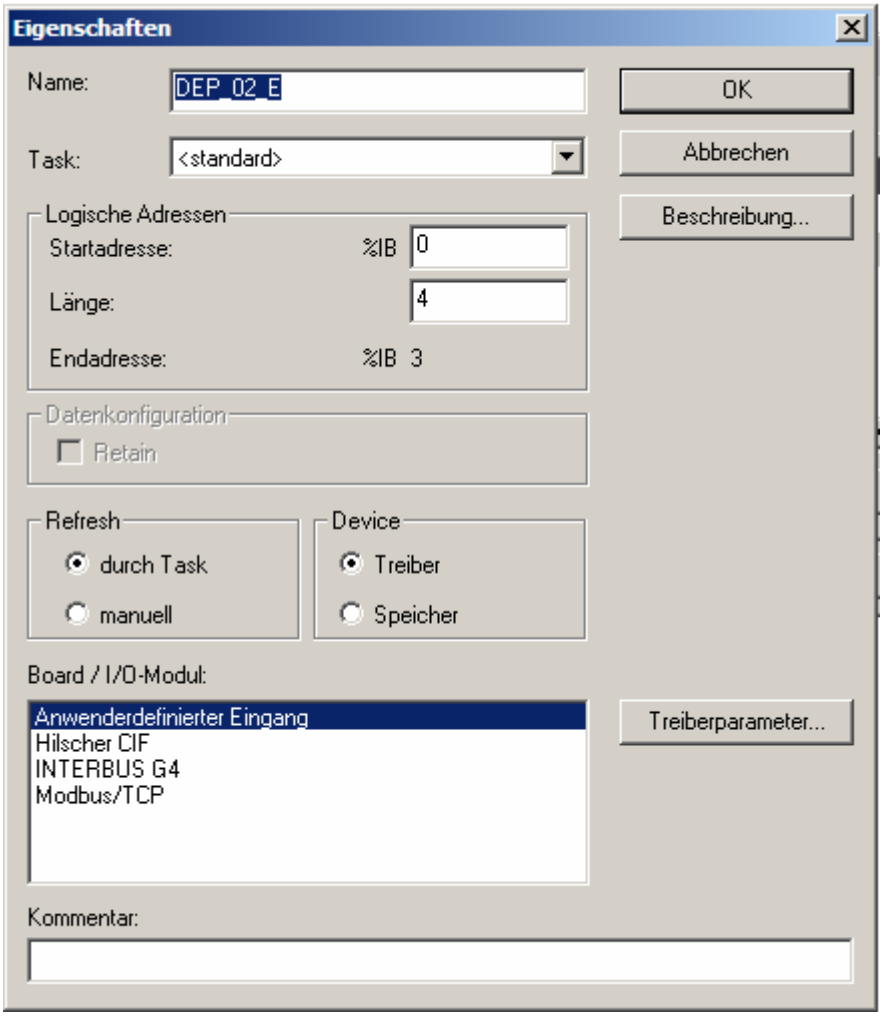
Beispiel: DEP_02_E für die Eingänge der DEP 512 auf Steckplatz 2

Die Längenangabe für die logischen Adressen ist baugruppenabhängig und können der folgenden Tabelle entnommen werden.

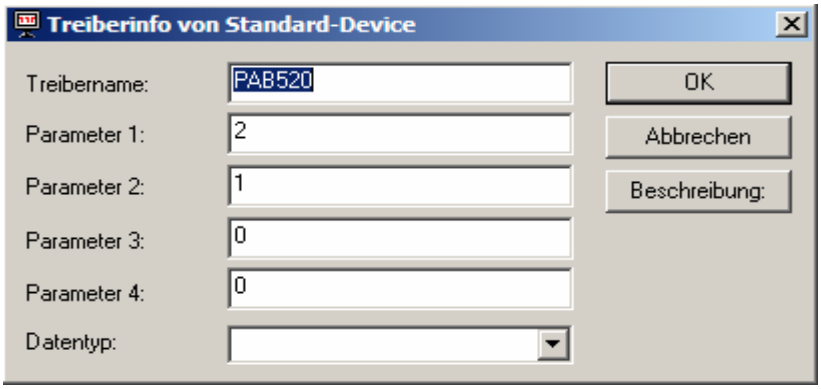
Baugruppenübersicht

Baugruppe	Ident-code	Anzahl Byte Eingabe	Ausgabe	Bemerkung
				Analoge Eingabe
ADU115 / ADU515	33	32		16 x 12 Bit +VZ
ADU116 / ADU516	3	32		16 x 10 Bit + VZ
				Analoge Ausgabe
DAU104	37	16	8	8 x 10 Bit + VZ / 4 x 10 Bit +VZ
DAU108 / DAU508	4		16	8 x 10 Bit + VZ
DAU109 / DAU509	52		16	8 x 10 Bit + VZ
				Digital Eingabe
DEP112 / DEP512	1	4		4 x 8 E
				Digital Ausgabe
DAP112 / DAP512	2		4	4 x 8A
DAP106 / DEA506	22		2	
				Digital kombinierte Baugruppen
DAP102	5	2	2	2 x 8E + 2 x 8A
DAP103 / DAP503	7	2	2	2 x 8E + 2 x 8A
				Kommunikationsbaugruppen
KOS140 / KOS540	91	180	180	Für Unterstationen und Unterzentralen mit SEAB 1F
KOS141 / KOS141	116	180	180	Für Unterstationen und Unterzentralen mit SEAB 1F
KOS560	94	128	128	Für Unterstationen und Unterzentralen mit IEC IEC 60870-5-101

4.2.1 Definition Standard I/O Module



Als Board / I/O-Modul ist immer die Einstellung "Anwenderdefinierter Eingang" zu wählen.
Rufen Sie anschließend über den Button "Treiberparameter…" das Menü zur Eingabe der Treiberinfo auf.

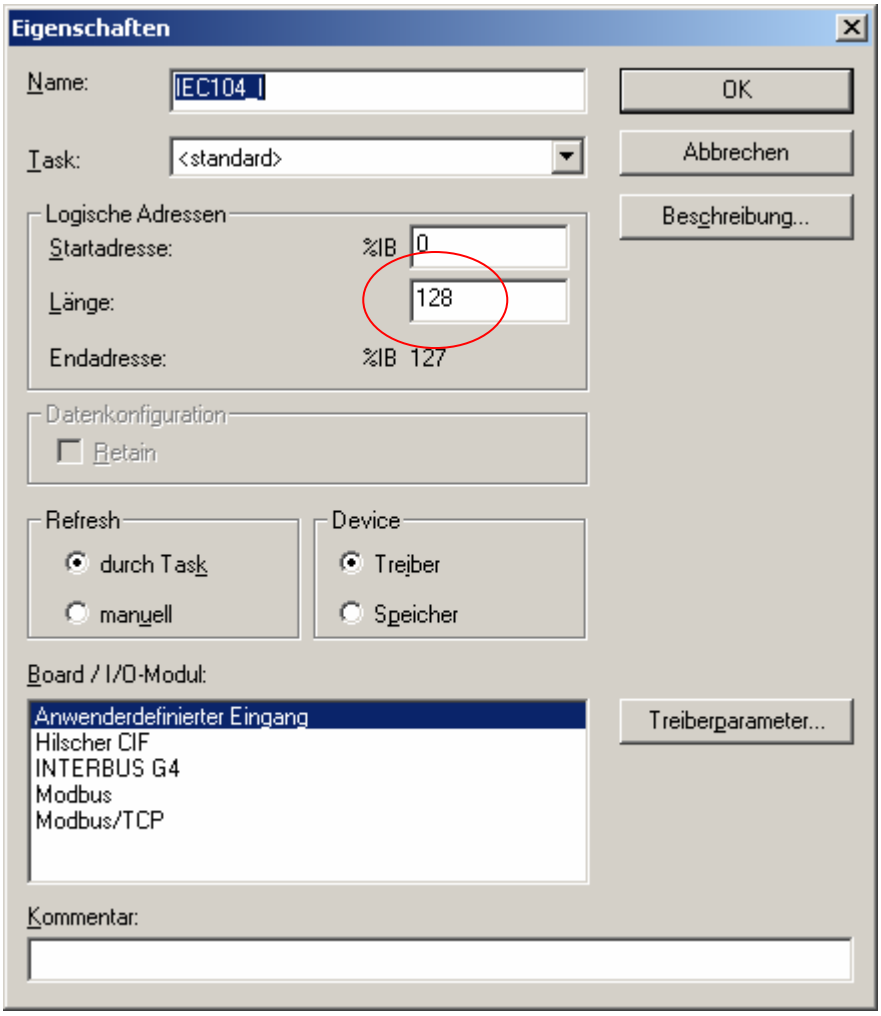


Eingabe **Treiberparameter**

- Als Treibername ist immer **PAB520** einzutragen.
- In Parameter 1 ist die Steckplatzadresse der Baugruppe einzutragen.
- In Parameter 2 wird der Identcode der Baugruppe eingetragen.

4.3 I/O Konfiguration für IEC 60870-5-101 oder IEC 60870-5-104

Die Bereiche für die Übergabe von Daten an die Kommunikationsschnittstelle der ALU 504/518 werden wie eine virtuelle I/O Baugruppe definiert. Es ist möglich mehrere dieser virtuellen Baugruppen zu projektieren.



Länge für Input
max. 512

Länge für Output
max. 1024

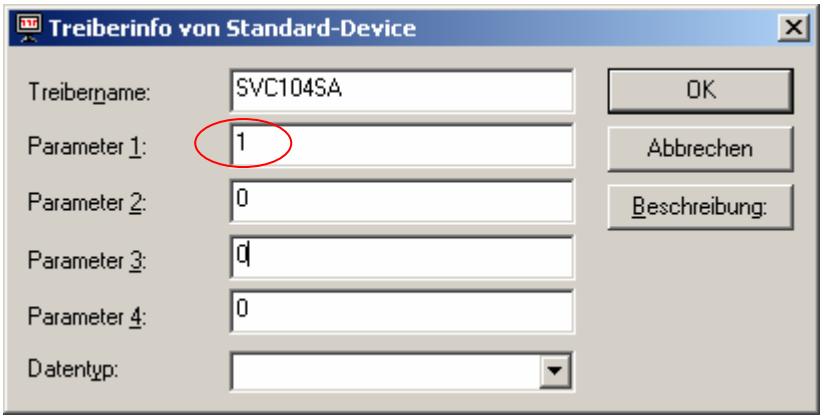
Der Name kann vom Anwender frei vergeben werden. Er kann aus bis zu 8 Zeichen bestehen und muss eindeutig sein. Es empfiehlt sich die Ein- und Ausgaberichtung durch ein I / O oder E / A zu kennzeichnen.

Anschließend werden über die **Treiberparameter** des **Anwenderdefinierten Eintrags** die Service-Routinen für IEC 60870-5-101 oder -104 zugeordnet.

Die ALU 504/518 kann bei entsprechender Projektierung (Datei PCOS_520.INI) gleichzeitig mehrere Service-Routinen bedienen.

Beispiel:

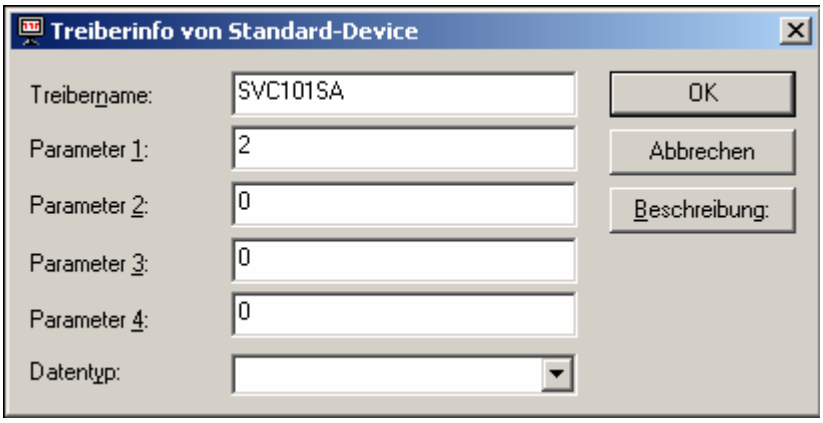
Service1 = IEC 60870-5-104 (Treiber = SVC104SA, TCP/IP Schnittstelle)
Service2 = IEC 60870-5-101 (Treiber = SVC101SA, serielle Schnittstelle COM 1)
Service3 = IEC 60870-5-101 (Treiber = SVC101SA, serielle Schnittstelle COM 2)



SVC104SA ist der Service für das IEC 60870-5-104 Protokoll.

Service Nummer in Parameter 1 muss identisch mit den Nummern in den Dateien SCV_COM? und SVC_RTU? sein.

Siehe auch Kap. 6



SVC101SA ist der Service für das IEC 60870-5-101 Protokoll.

Service Nummer in Parameter 1 muss identisch mit den Nummern in den Dateien SCV_COM? und SVC_RTU? sein.

Siehe auch Kap. 6

Achtung:

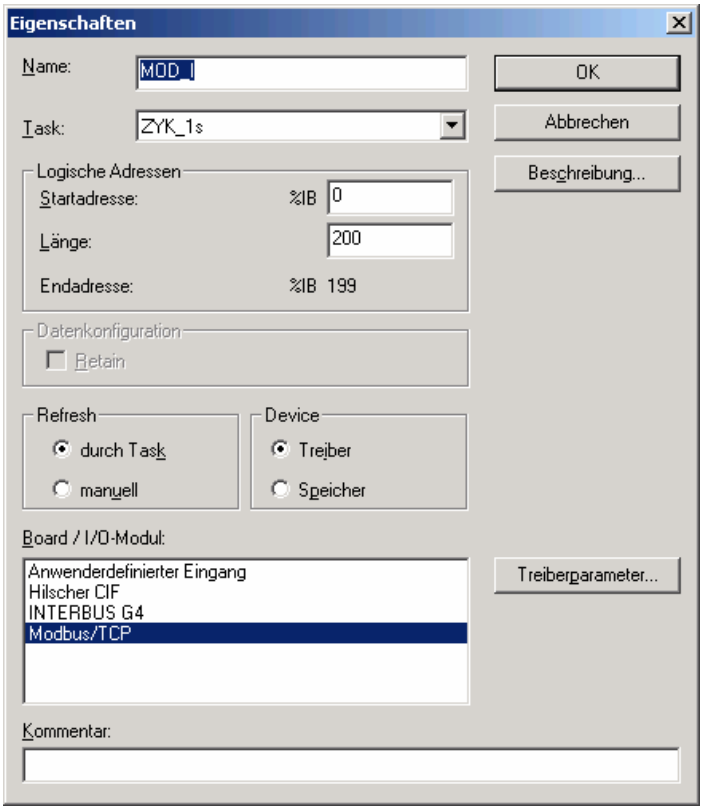
Achten Sie unbedingt darauf, dass die in der I/O Konfiguration projektierten Service-Routinen in der PCOS-A520 freigegeben sind, bevor Sie das Projekt an die ALU senden. Wenn diese Routinen nicht freigegeben sind, wird der Ladevorgang mit Fehler abgebrochen.

Weiterführende Informationen siehe Kapitel 6.1 und Kapitel 6.3.

4.4 I/O Konfiguration für Modbus TCP Client

Für diese Anbindung muss ein spezieller Modbus/TCP-Treiber für MULTIPROG installiert werden. Dieser ist auf Anfrage bei OHP erhältlich. Nach der Installation wird der Treiber im Auswahlfenster "Board / I/O-Modul" angezeigt.

Die Projektierung wird anhand des INPUT Dialogs der I/O-Konfiguration erläutert.



Task:

Um feste Abfrageintervalle für den Modbus zu erreichen, definieren Sie für den Aufruf des Treibers eine zyklische Task. Falls in Ihrer Applikation keine zyklischen Programmaufrufe benötigt werden, können sie hierfür auch ein leeres Dummy-Programm erstellen.

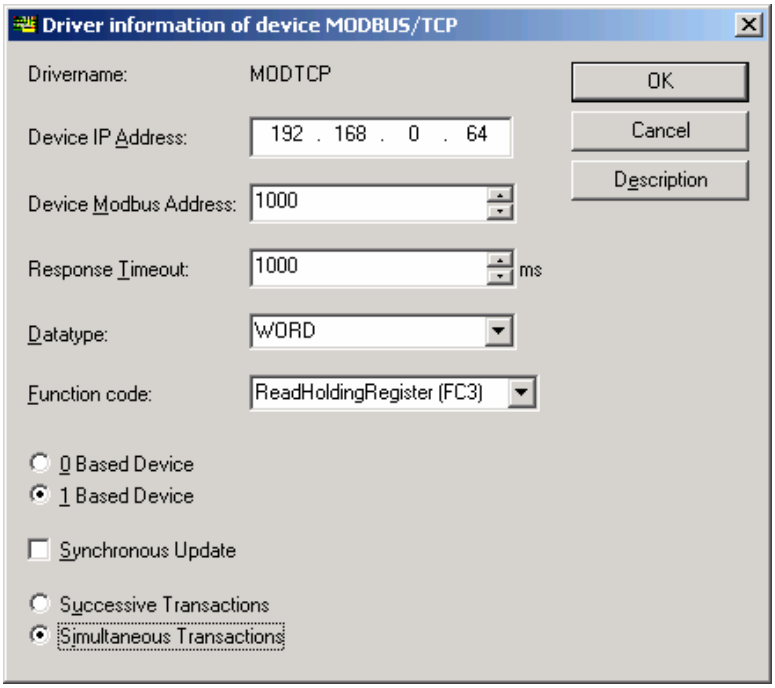
Logische Adressen:

In diesem Abfragedialog geben Sie das 1. IB an, ab dem die übertragenen Daten abgelegt werden.

Länge:

Die Länge der abgefragten Daten wird in Anzahl Byte angegeben. Im obigen Beispiel werden 200 Byte = 100 WORD bzw. Register aus der angekoppelten Station abgefragt.

Über den Dialog "Treiberparameter…" wird festgelegt, welche Register aus der Station ausgelesen werden.



Device IP Adresse:

In das Feld "Device IP Address" tragen Sie die IP-Adresse der angekoppelten Station ein.

Device Modbus Address:

In das Feld "Device Modbus Address" tragen Sie die Adresse des 1. Registers ein, ab dem ausgelesen werden soll.

In der Beispiel-Projektierung werden ab Register 1000 (401000, MW 1000 etc.) 100 Register aus der angekoppelten Station abgefragt und in MULTIPROG im Bereich IB0 … IB199 abgelegt.

Funktion code:

Hier können Sie einstellen, ob Sie auf Register (WORD), Coils (BIT) oder direkt auf die I/O der angeschlossenen Station zugreifen wollen.

Der ausgewählte Funktionscode muss durch die angeschlossene Station unterstützt werden.

Mit **Based Device** 0 oder 1 geben Sie an, ob die angekoppelte Station in ihrer Zählweise mit Register 0 oder 1 beginnt.

Über die Definition mehrere I/O-Gruppen können Sie verschiedene Bereiche aus der angekoppelten Station abfragen. Die Bereiche können nacheinander (**Successive Transactions**) oder parallel (**Simultaneous Transactions**) abgefragt werden.

Nacheinender heißt, dass das Abfragetelegramm für den nächsten Bereich erst dann abgeschickt wird, wenn die Antwort auf die zuvor gesendete Abfrage eingetroffen ist, oder die unter **Response Timeout** projektierte Zeit abgelaufen ist.

Parallel heißt, dass mehrer Abfragetelegramme hintereinander geschickt werden können, unabhängig davon, ob eine Antwort eingetroffen ist. Die Datenübertragung ist also asynchron.

Für die Projektierung von Ausgaben über den Modbus/TCP gehen Sie in der gleichen Weise vor wie für die Eingaben. Benutzen Sie hierfür den OUTPUT Dialog der I/O-Konfiguration.

4.5 Projektierung des Shared-Memory-Bereichs für Modbus RS232-Slave / TCP-Server

Die ALU 504/518 stellt insgesamt einen Bereich von 16 kB (MB3.0000 bis MB3.16383) für das Shared-Memory zur Verfügung. Dieser Bereich oder Teile davon können für die Modbus RS232-Slave oder TCP-Server Schnittstelle reserviert werden.

Die Definition des Bereichs erfolgt über INI-Dateien, die auf der CF-Karte abgelegt werden. Diese Dateien PCOS_A520.INI und MODBUS_SERVER.INI befinden sich im Verzeichnis PCOS_OHP.

Weiterführende Informationen siehe Kap. 6.4.

5 Systemmerker

Für die maximal 10 Steckplätze im Zentralbaugruppenträger einer @250 (DTA113) sind im Systemmerkerbereich mehrere Doppelwort für die Fehler- und Statusanzeige definiert. Statusinformationen werden nur für Steckplätze geliefert, die in der I/O-Konfiguration definiert sind.

Bei einfachen E/A-Baugruppen werden nur die Systemmerker "Baugruppenausfall" und "Baugruppenstörung" bedient. Für Expertenbaugruppen (z.B. KOS xxx) und komplexere E/A-Module (z.B. DEZ 161) werden zusätzlich noch 4 Statusbyte zur Verfügung gestellt.

Der Systemmerkerbereich hat eine feste Struktur. Merkerbit und Merkerbyte für Steckplätze die nicht vorhanden bzw. nicht bestückt sind, sind immer 0.

Hinweis für Programmierer die A250-Systeme umrüsten auf @250.
Das Systemmerkerdoppelwort "Baugruppenausfall" %MD1.3000 entspricht den SM(tn).1 in AKF25.
Das Systemmerkerdoppelwort "Baugruppenstörung" %MD1.3004 entspricht den SM(tn).2 in AKF25.

Die Systemmerker können als Doppelwort, Byte oder Bit angesprochen werden.

Um einen Sammelfehler für Baugruppenausfall zu erzeugen, kann z.B. das %MD1.3000 auf ungleich NULL geprüft werden. Für die Prüfung einzelner Steckplätze verwendet man Sinnvollerweise die Merkerbit %MX1.3000.1, %MX1.3000.2 usw.

Die Definition entnehmen Sie bitte den folgenden Tabellen:

5.1 Systemmerker Projekt

%MB1.332	Bootprojekt vorhanden
%MB1.333	Projektquellen vorhanden
%MB1.334	Projekt im RAM
%MD1.336	CRC des Bootprojektes
%MD1.340	CRC Projektquellen
%MD1.344	CRC des Projektes im RAM
%MB1.348	Name des Projektes: STRING
%MB1.380	Name der Konfiguration: STRING
%MB1.412	Name der Ressource: STRING

5.2 Systemmerker Board

%MD1.2932	BOOT COUNTER (ab PCos 3.52)
%MD1.2936	reserviert - CPU Temperatur Sensor nur @250
%MD1.2940	reserviert - Reserve Sensor CPU Temperatur nur @250
%MD1.2944	CPU Temperatur Sensor nur @250
%MD1.2948	Status Sensor CPU Temperatur nur @250

5.3 Systemmerker Service

%MD1.2952	%MB1.2952	Reserviert
	%MB1.2953	Reserviert
	%MB1.2954	Reserviert
	%MB1.2955	Reserviert
%MD1.2956	%MB1.2956	Statusbyte Service Sammelmerker
	%MB1.2957	Statusbyte Service Sammelmerker
	%MB1.2958	Statusbyte Service Sammelmerker
	%MB1.2959	Statusbyte Service Sammelmerker
%MD1.2960	%MB1.2960	Statusbyte Service 1
	%MB1.2961	Statusbyte Service 1
	%MB1.2962	Statusbyte Service 1
	%MB1.2963	Statusbyte Service 1
.	.	.
.	.	.
.	.	.
%MD1.2992	%MB1.2992	Statusbyte Service 9
	%MB1.2993	Statusbyte Service 9
	%MB1.2994	Statusbyte Service 9
	%MB1.2995	Statusbyte Service 9

Statusbyte Service Sammelmerker MD1.2956:

Bitposition entspricht Sammelfehler eines System-Service SVC0 bis SVC9 (SVC0 ist immer für internen System-Service reserviert).

SVC1 bis SVC 9 können derzeit für IEC 60870-5-101 oder IEC 60870-5-104 Kommunikation projektiert werden.

Sammelfehler-Bit0 von MD1.2956 reserviert
Sammelfehler-Bit1 von MD1.2956 für Service 1
Sammelfehler-Bit2 von MD1.2956 für Service 2
usw.

Statusbyte1 Service n:

Bit 0	nicht definiert
Bit 1	1 = Teilnehmer hat keine Verbindung, 0 = Verbindung OK (Zykl. Timeout)
Bit 2	1 = Bus keine Verbindung, 0 = Bus Ok (Bus Timeout)
Bit 3…7	nicht definiert

Die Statusbyte 2…4 werden derzeit nicht verwendet.

5.4 Systemmerker Steckplatz in I/O Konfiguration belegt

%MD1.2996	%MB1.2996	%MX1.2996.0	ALU (Platz 1) - reserviert	
		%MX1.2996.1	E/A Karte (Platz 2)	Merker für Platz 1 bis 10
		%MX1.2996.2	E/A Karte (Platz 3)	
		%MX1.2996.3	E/A Karte (Platz 4)	0 … nicht projektiert
		%MX1.2996.4	E/A Karte (Platz 5)	1 … belegt
		%MX1.2996.5	E/A Karte (Platz 6)	
		%MX1.2996.6	E/A Karte (Platz 7)	
		%MX1.2996.7	E/A Karte (Platz 8)	
	%MB1.2997	%MX1.2997.0	E/A Karte (Platz 9)	
		%MX1.2997.1	E/A Karte (Platz 10)	
	%MB1.2998		Reserviert	
	%MB1.2999		Reserviert	

Diese Merker zeigen an, ob ein Steckplatz in der I/O Konfiguration belegt wurde. Ist ein Platz projektiert, wird das entsprechende Bit auf 1 gesetzt.

5.5 Systemmerker Baugruppenausfall

%MD1.3000	%MB1.3000	%MX1.3000.0	ALU (Platz 1) - reserviert	
		%MX1.3000.1	E/A Karte (Platz 2)	Merker für Karte 1 bis 10
		%MX1.3000.2	E/A Karte (Platz 3)	
		%MX1.3000.3	E/A Karte (Platz 4)	0 … OK
		%MX1.3000.4	E/A Karte (Platz 5)	1 … **Ausgefallen** – nur projektierte Karten
		%MX1.3000.5	E/A Karte (Platz 6)	
		%MX1.3000.6	E/A Karte (Platz 7)	
		%MX1.3000.7	E/A Karte (Platz 8)	
	%MB1.3001	%MX1.3001.0	E/A Karte (Platz 9)	
		%MX1.3001.1	E/A Karte (Platz 10)	
	%MB1.3002		Reserviert	
	%MB1.3003		Reserviert	

Ist ein Steckplatz in der I/O Konfiguration projektiert und nicht bestückt bzw. die Baugruppe defekt, wird diese als ausgefallen gemeldet.

5.6 Systemmerker Baugruppenstörung

%MD1.3004	%MB1.3004	%MX1.3004.0	ALU (Platz 1) - reserviert	
		%MX1.3004.1	E/A Karte (Platz 2)	Merker für Karte 1 bis 20
		%MX1.3004.2	E/A Karte (Platz 3)	
		%MX1.3004.3	E/A Karte (Platz 4)	0 … OK
		%MX1.3004.4	E/A Karte (Platz 5)	1 … **Störung**
		%MX1.3004.5	E/A Karte (Platz 6)	
		%MX1.3004.6	E/A Karte (Platz 7)	
		%MX1.3004.7	E/A Karte (Platz 8)	
	%MB1.3005	%MX1.3005.0	E/A Karte (Platz 9)	
		%MX1.3005.1	E/A Karte (Platz 10)	
	%MB1.3006		Reserviert	
	%MB1.3007		Reserviert	

Einige Baugruppen wie z.B. die KOSxxx können Störungen melden. Der Systemmerker Baugruppenstörung dient als Sammelmerker für diese Störungen. Die genaue Störungsursache wird über die Statusbyte (siehe Kap. 5.7) bekannt gegeben.

5.7 Systemmerker Statusbyte

%MD1.3008	%MB1.3008	Statusbyte E/A Karte – Platz 1
	%MB1.3009	Statusbyte E/A Karte – Platz 1
	%MB1.3010	Statusbyte E/A Karte – Platz 1
	%MB1.3011	Statusbyte E/A Karte – Platz 1
%MD1.3012	%MB1.3012	Statusbyte E/A Karte – Platz 2
	%MB1.3013	Statusbyte E/A Karte – Platz 2
	%MB1.3014	Statusbyte E/A Karte – Platz 2
	%MB1.3015	Statusbyte E/A Karte – Platz 2
.	.	.
.	.	.
.	.	.
%MD1.3044	%MB1.3044	Statusbyte E/A Karte – Platz 10
	%MB1.3045	Statusbyte E/A Karte – Platz 10
	%MB1.3046	Statusbyte E/A Karte – Platz 10
	%MB1.3047	Statusbyte E/A Karte – Platz 10

5.7.1 Definition Statusbyte KOS140

1. Statusbyte

Bit 0	Steckplatzfehler KOS
Bit 1	SEAB Verkehr gestört AWD: Anz. Wahlversuche erfolglos
Bit 2	Min. Impuls fehlt
Bit 3	Zeit fehlt
Bit 4	Keine Parameter
Bit 5	KOS hat falsche Teilnehmernummer
Bit 6	
Bit 7	Ringpuffer voll (Slave)

2. Statusbyte, 3. Statusbyte, 4. Statusbyte – nur intern

5.7.2 Uhrzeitübergabe durch KOS 140

Die KOS übergibt in 8 aufeinander folgenden Byte die Uhrzeit, sofern diese synchronisiert ist. Die Synchronisation kann über SEAB-1F Zeittelegramme oder über DCF-Funksignal erfolgen. Diese Uhrzeitübergabe kann wiederum zum Synchronisieren der CPU-Uhr verwendet werden.

Der KOS Übergabebereich besteht aus 180 Byte, die Uhrzeit wird in die letzten 8 Byte eingetragen.

Die 8 Byte sind wie folgt definiert:

Byte	Bezeichnung	Codierung
173	Jahr	BCD
174	Monat	BCD
175	Wochentag	BCD
176	Tag	BCD
177	Stunde	BCD
178	Minute	BCD
179	Sonderzeichen	1. Bit = Umschaltung auf Reserveante 2. Bit = Ankündigung SZ/WZ Umschaltung 3. Bit = Sommerzeit 4. Bit = Winterzeit 5. Bit = Schaltsekunde
180	Sekunde	BCD

Die Angabe der Byte ist immer relativ zum projektierten %IB Bereich in der IO-Konfiguration zu sehen.

5.7.3 Definition Statusbyte KOS141

1. Statusbyte

Bit 0	Falsche Teilnehmernummer
Bit 1	Minutenimpuls fehlt (DCF)
Bit 2	Pollstopp Linie 1
Bit 3	Pollstopp Linie 2
Bit 4	
Bit 5	
Bit 6	
Bit 7	

2. Statusbyte, 3. Statusbyte, 4. Statusbyte – nur intern

Pollstopp bedeutet, dass die bei den Übertragungsparametern der KOS vorgegebenen Überwachungszeit für den Pollzyklus abgelaufen ist (die Zentrale pollt nicht mehr). Das Bit wird zurückgesetzt, wenn die Zentrale den Pollbetrieb wieder aufnimmt.

5.7.4 Definition Statusbyte Modbus

1. Statusbyte

Bit 0	Schnittstelle: 1 = Bereit, 0 = Fehler
Bit 1	Schnittstelle: 1 = Läuft, 0 = Stopp
Bit 2	Reserve
Bit 3	Reserve
Bit 4	COM-Task: 1 = Bereit, 0 = Fehler
Bit 5	Reserve
Bit 6	Reserve
Bit 7	Fest auf 1

2. Statusbyte

Bit 0	Kommunikationsstatus: 1 = Verbunden, 0 = nicht verbunden
Bit 1	Reserve
Bit 2	Reserve
Bit 3	Reserve
Bit 4	Reserve
Bit 5	Reserve
Bit 6	Reserve
Bit 7	Reserve

3. Statusbyte

Nummer der Funktion, die den letzten Fehler verursacht hat.

4. Statusbyte

Modbus Fehlercode

1	Unbekannter bzw. unzulässiger Funktionscode
2	Abgefragte Adresse / Bereich nicht bekannt
3	Illegaler Abfragewert (abhängig vom Abfragetyp)

5. Statusbyte

Fortlaufender Fehler-Zähler mit Umschlag bei 255.

6. Statusbyte

Fortlaufender Antwort-Zähler mit Umschlag bei 255.

Die Speicher-Adresse und Anzahl der Statusbyte wird in der Datei MODBUS_SERVER.INI eingestellt. Die Statusbyte 3 …6 sind optional.

Weiterführende Informationen siehe Kapitel 6.4.1.

6 Projektierung der Initialisierungsdateien

6.1 Einstellen der IP-Adresse auf der 504/518

Im ROOT-Verzeichnis der CF-Karte befindet sich die Datei ALU520.INI. Diese Datei beinhaltet die Einstellungen bezüglich der IP-Adresse.

```
#ALU520:  (GEN)

[IP]
ADDRESS          = 192.168.0.52
NETMASK          = 255.255.255.0
DHCP             = 0

[IP-IF2]

ADDRESS          = 192.168.0.51
NETMASK          = 255.255.255.0
DHCP             = 0

[FTP]
ENABLE           = 1

[TELNET]
ENABLE           = 1

[TIME-GPS-NMEA]
ENABLE           = FALSE
PORT             = 2
BAUD             = 4800          # darf nicht geändert werden
REFRESH          = 60            #  in Minuten
```

6.1.1 Einstellen der IP-Adresse auf der ALU504/518

ADRESSE und NETMASK müssen immer vorgegeben werden. Wenn Sie die ALU über ein Gateway betreiben möchten, können Sie die Datei wie folgt erweitern.

Beispiel:

```
ADDRESS= 192.168.101.101
NETMASK= 255.255.255.0
GATEWAY= 192.168.101.1
```

Sofern Sie die Datei über einen FTP Zugriff geändert haben, müssen Sie die ALU Aus- und Einschalten, damit die neuen Daten übernommen werden.

6.1.2 Zugriff über FTP oder TELNET

Die Einträge [FTP] und [TELNET] ermöglichen den Zugriff auf die ALU mit den entsprechenden Diensten. Wenn Sie den Zugriff über diese Dienste nicht erlauben wollen, setzten Sie ENABLE =0.

Achtung: Wenn Sie den FTP-Zugriff deaktivieren, können Sie die INI-Dateien auf der ALU nur noch über einen Card-Reader ändern.

6.1.3 Zeitsynchronisation über GPS

Parameter unter [TIME-GPS-NMEA]

Wird der Parameter **ENABLE** auf TRUE gesetzt, kann die Zeitsynchronisation der ALU über einen GPS-Empfänger erfolgen.

PORT gibt die COM-Schnittstelle (1 oder 2) an, an den der Empfänger angeschlossen wird.

BAUD gibt die Übertragungsgeschwindigkeit an, mit der der Empfänger die Daten an die ALU überträgt. Dieser Eintrag darf nicht verändert werden.

REFRESH gibt das Zeitintervall in Minuten (1…60) an, in dem die Systemuhr durch den GPS-Empfänger synchronisiert wird. D.h. innerhalb dieser Zeit treffen zwar mehrere Zeitinformation vom GPS-Empfänger ein, jedoch wird die Systemuhr erst nach Ablauf dieser Zeit durch das nächste gültige Zeittelegramm synchronisiert. Nach jeder erfolgreichen Synchronisation wird der Refresh-Timer wieder neu gestartet.

Siehe auch Kapitel 8.3.

6.2 Freigabe der Service-Routinen und Interfaces

Die einzelnen Service-Routinen und Schnittstellen werden über die Datei PCOS_A520.INI frei gegeben. Die Datei befindet sich im Verzeichnis PCOS_OHP auf der CF-Karte.

☞ **Hinweis:** Ab Firmware-Version 1.70 werden geänderte Dateinamen verwendet (PCOSA520.INI)

Bei Auslieferung der CF-Karte sind alle Einträge inaktiv. Durch Entfernen des Doppelschrägstrichs // wird ein Eintrag freigegeben.

Achtung: Schalten Sie in Ihrem Projekt nur die Service-Routinen und Schnittstellen (Interface) frei, die Sie tatsächlich benötigen. Anhand der Einstellungen in dieser Datei werden beim Hochlauf der ALU 520 die Treiberdateien (DLLs) in den Speicher geladen. Jeder freigegebene Treiber belegt Speicherplatz, auch dann, wenn er nicht verwendete wird.

```
#=================================================================
#
#       ALU520 PCOS Einstellungen
#       ALU520 PCOS settings
#
#=================================================================
[PCOS-SYSTEM]                                   #Speicherbereich für Shared-Memory
SharedMemoryAddr=1000                           #Startadresse ProConOs  %MB3.1000
SharedMemorySize=600                            #Anzahl Bytes in Summe     bis %MB3.1599
#=================================================================
#
#       MODBUS Server Einstellungen: Speicher, Schnittstellen
#       max. 4 Schnittstellen möglich
#
#       MMODBUS server settings: memory, interface
#       max. 4 interface are possible
#
#=================================================================
[MODBUS-SERVER]                                 #Speicherbereich für Modbus
//SharedMemoryOffset=200                        #Startadresse
//SharedMemorySize=308                          #Anzahl Bytes in Summe
//Interface1= SSERIAL                           #Slave Serial
//Interface2= STCP_IP                           #Modbus TCP-Server

#=================================================================
#
#    Service-Routinen für IEC 60870-5-101 oder IEC 60870-5-104
#
#=================================================================
[SERVICE]                                       #Protokoll IEC 60870-5-104  (TCP/IP)
//Service1= SVC104SA                            #Protokoll IEC 60870-5-101  (seriell)
//Service2= SVC101SA
```

Beispiel:

Mit der auf der CF-Karte enthaltenen Beispielprojektierung kann durch Freigabe von Interface1 und Service2 z.B. eine Datenübertragung nach IEC 60870-5-101 über die COM1 und eine Ankopplung an einen Modbus-Master über die COM2 der ALU 520 realisiert werden.

6.2.1 Definition von Shared Memory

Die Startadresse des Shared Memory kann bei 0 (%MB3.000) beginnen. Bitte achten Sie darauf als Startadresse immer gerade Adressen zu verwenden (0, 2, 4, 200, 1000 etc.)

Die Länge kann bis maximal 16384 vorgegeben werden. Geben Sie auch bei der Länge eine gerade Zahl ein.

Im Beispiel steht dem Anwenderprogramm (AWP) ein Shared Memory Bereich von %MB3.1000 bis %MB3.1599 zur Verfügung.

Dieser kann z.B dafür benutzt werden, um überlappende Variablen oder Strukturen zu definieren.

Beispiel:

Variable A	BYTE	%MB3.1200
Variable A1	BOOL	%MX3.1200.0
.		
.		
Variable A8	BOOL	%MX3.1200.7

Der gesamte Shared Memory Bereich oder Teile davon können für eine Modbus-Ankopplung verwendet werden.

6.2.2 Modbus RS232

Werden die Einträge für den Modbus durch Löschen der // freigegeben, kann im Beispiel der Bereich %MB3.1200 bis %MB3.1507 über die Modbus RS232-Schnittstelle beschrieben und gelesen werden.

[MODBUS-SERVER]

SharedMemoryOffset=200
SharedMemorySize=308

Interface1= **SSERIAL**

Sie können auch den kompletten Shared Memory Bereich für die Modbus-Übertragung zur Verfügung stellen. Im Beispiel wäre das mit folgenden Einstellungen möglich:

[MODBUS-SERVER]

SharedMemoryOffset=0
SharedMemorySize=600

Interface1= **SSERIAL**

Über die Datei MODBUS_SERVER.INI wird dieser Bereich auf die einzelne Coils und Register zugewiesen.

Weiterführende Informationen siehe Kapitel 6.4.

6.2.3 Modbus TCP-Server

Werden die Einträge für den Modbus TCP-Server durch Löschen der // beim Interface2 freigegeben, kann im Beispiel der Bereich %MB3.1200 bis %MB3.1507 über Modbus TCP beschrieben und gelesen werden.

[MODBUS-SERVER]

SharedMemoryOffset=200
SharedMemorySize=308

Interface2= **STCP_IP**

Sie können auch den kompletten Shared Memory Bereich für die Modbus-Übertragung zur Verfügung stellen. Im Beispeil wäre das mit folgenden Einstellungen möglich:

[MODBUS-SERVER]

SharedMemoryOffset=0
SharedMemorySize=600

Interface2= **STCP_IP**

Über die Datei MODBUS_SERVER.INI wird dieser Bereich auf die einzelne Coils und Register zugewiesen.

Weiterführende Informationen siehe Kapitel 6.4.

6.2.4 Service-Routinen

Soll eine Kommunikation über IEC 60870-5-101 und/oder IEC 60870-5-104 aufgebaut werden, sind die entsprechenden Service-Routinen freizugeben. Es ist möglich mehrere Routinen gleichzeitig zu starten.

Die Einstellung
Service1= SVC104SA (zugehörige Parameter-Dateien SVC_COM1.INI und SVC_RTU1.INI)
Service2= SVC101SA (zugehörige Parameter-Dateien SVC_COM2.INI und SVC_RTU2.INI)

ermöglicht die gleichzeitige Übertragung von Daten über TCP/IP und eine serielle Ankopplung.

Die Einstellung
Service1= SVC101SA (zugehörige Parameter-Dateien SVC_COM1.INI und SVC_RTU1.INI)
Service2= SVC101SA (zugehörige Parameter-Dateien SVC_COM2.INI und SVC_RTU2.INI)

ermöglicht die gleichzeitige Übertragung der Daten über die beiden seriellen Schnittstellen der ALU 504/518 mit IEC 60870-5-101 Protokoll.

Anmerkung: Bei Auslieferung der CF-Karte ist die Datei SVC_COM1.INI für das Protokoll IEC 60870-5-104 vorbereitet. Wenn wie oben beschrieben beide Service-Routinen mit dem seriellen Protokoll arbeiten sollen, kopieren Sie den Inhalt der Datei SVC_COM2.INI in die SVC_COM1.INI und ändern Sie die Einstellungen entsprechend Ihrer Anwendung ab. Achten sie insbesondere darauf, dass die Einstellung der Ports in den beiden Dateien unterschiedlich ist.

Hinweis: Ab Firmware-Version 1.70 werden geänderte Dateinamen verwendet (SVCCOMn.INI)

6.3 IEC 60870-5-101 oder -104 Kommunikation

6.3.1 Allgemeines

Für die einzelnen Übertragungsprotokolle können auf der ALU 504/518 bis zu 8 Service-Routinen gestartet werden. Jeder Service-Routine werden Parameterdateien für die Kommunikation (SVC_COM?.INI) und die Definition der PV-Verarbeitung (SVC_RTU?.INI) zugeordnet.

Es sind die Protokolle IEC 60870-5-101 und IEC 60870-5-104 mit Slave-Funktionalität verfügbar.

Die Parameterdateien SVC_COM?.INI und SVC_RTU?.INI befinden sich im Verzeichnis PCOS-OHP auf der CF-Karte.

Hinweis: Ab Firmware-Version 1.70 werden geänderte Dateinamen verwendet (SVCCOMn.INI und SVCRTUn.INI)

Anmerkung: Die einzelnen Parameter der auf der CF-Card enthaltenen Dateien können von dieser Darstellung abweichende Zuweisungen haben.

Ändern Sie die Einträge entsprechend Ihrer Aufgabenstellung und speichern Sie die Dateien anschließend wieder auf der CF-Card.

Hinweis: Wird einem Eintrag ein Doppel-Schrägstrich "//" vorangestellt, wird die entsprechende Zeile nicht interpretiert. Dies kommt einem Löschen der Zeile gleich und der entsprechende Parameter ist nicht aktiviert. Nicht benötigte Parameter können gelöscht oder deaktiviert werden.
Die Defaulteinstellung für gelöschte oder deaktivierte Parameter ist 0, FALSE oder OFF.

Der Eintrag TRUE entspricht ON, der Eintrag FALSE entspricht OFF. Es können jeweils beide Varianten der Bezeichnungen verwendet werden.

6.3.2 Parameterdatei für die Kommunikationsprojektierung IEC 60870-5-104 (SVC_COM?)

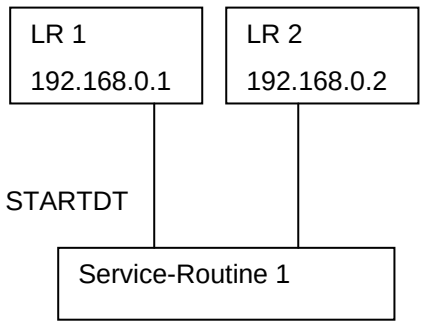
[IEC-5-104]	
PORT= 2404	Portnummer **Darf nicht geändert werden!**
KVAL= 12	Maximale Differenz zwischen Anzahl Empfangsfolgen und Sendefolgen
WVAL= 8	Späteste Quittierung nach Empfang von w APDU im I-Format
T1= 15	Zeitüberwachung für gesendete APDU oder Test-APDU
T2= 10	Zeitüberwachung für Quittierungen, falls keine Daten übertragen werden
T3= 40	Zeitüberwachung für gesendete Testframes

Über eine Service-Routine können maximal zwei IEC 60870-5-104 Verbindungen aufgebaut werden. Jedoch darf nur bei **einer** Verbindung das STARTDT aktiviert werden. Bei der 2. Verbindung werden nur die Testframes (TESTFR) bestätigt.

Sie können die IP-Adresse des Kommunikationspartners eintragen (max. 2). Damit wird sichergestellt, dass die ALU520 eine TCP/IP-Verbindung nur zu dieser Adresse zulässt. Verbindungsversuche von anderen Adressen werden nicht bestätigt.

Wird keine Adresse eingetragen, kann über jede beliebige IP-Adresse im gleichen Subnetz eine Verbindung etabliert werden.

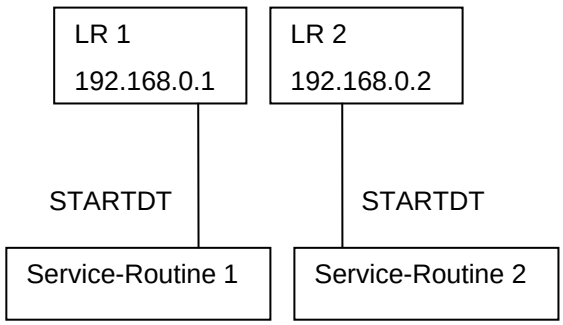
☞ **Hinweis:** Werden für eine redundante Datenübertragung zwei Verbindungen, mit aktiviertem STARTDT benötigt, müssen zwei Service-Routinen projektiert werden.



Daten werden nur mit LR1 ausgetauscht.

Bei LR2 werden nur TESTFR beantwortet.

Ein Wechsel kann nur durch STOPDT bei LR1 und anschließendem STARTDT bei LR2 veranlasst werden



Daten werden mit beiden LR ausgetauscht

6.3.3 Parameterdatei für die Kommunikationsprojektierung IEC 60870-5-101 (SVC_COM?)

Öffnen Sie die Datei ATX_COM.INI die Sie auf Ihrem Rechner gespeichert haben mit dem Editor.

Sie sehen jetzt folgende Einträge:

[IEC-5-101]	
AWD= FALSE	TRUE = AWD-Betrieb, FALSE = Standleitungsbetrieb
PORT= 1	Port =1 oder 2 entsprich COM - Schnittstelle.
BAUDRATE= 9600	Übertragungsrate zum Leitrechner
PARITY= EVEN	Parität NO / EVEN (bei Standleitung immer EVEN)
DATA=8	Anzahl Datenbits **Darf nicht geändert werden!**
STOPBIT=1	Stoppbit **Darf nicht geändert werden!**
TVS2=25	Vorlaufzeit S2 [ms] (nur relevant bei Standleitung)
TNS2=7	Nachlaufzeit S2 [ms] (nur relevant bei Standleitung)
TVM2=40	Vorlaufüberwachung M2 [ms] (nur relevant bei Standleitung)
TNM2=150	Nachlaufüberwachung M2 [ms] (nur relevant bei Standleitung)
PAUSE=27	Pausenzeit [ms]
BUS_TOUT= 2	Busüberwachung (Slave) in n*[100ms]
POLL_TOUT= 60	Überwachung Pollzyklus (Slave) in n*[100ms]
S2SIGNAL= OFF	S2-Signal (RTS) ein- oder ausschalten
M2SIGNAL= OFF	M5/M2-Überwachung (DCD/CTS) ein- oder ausschalten
CONF_SC= OFF	ON= Empfangsbestätigung mit Einzelzeichen E5, OFF = Kurztelegramm
AVS= 3	Adresse der Verbindungsschicht
OKTETT_AVS= 2	Anzahl Oktett der AVS (0 bis 2)
[AWD101-MODE]	!! AT Befehle ASCII, max. 8 Zeichen !!
AT_INIT= ATZ0	Abgespeichertes Profil in Modem laden
AT_HANGUP= ATH	AT-Kommando für Auflegen
AT_DIAL= ATD	AT-Kommando für Wählen
AT_PICK_UP= ATA	AT-Kommando für Abheben
# Modem RESPONSE/Rückmeldung Dezimal	

//RSP_OK= 0
//RSP_CONNECT= 1
//RSP_RING= 2
//RSP_NOCARRIER= 3
//RSP_ERROR= 4
//RSP_NODIAL= 5
//RSP_BUSY= 6
//RSP_NOANSWER=7
	Hier sind die bei den Modems üblichen Werte für die Rückmeldungen angegeben. Sollte das von Ihnen verwendete Modem davon abweichende Werte verwenden, löschen Sie vor dem entsprechenden Eintrag den Doppel-Slash "//" und geben Sie den verwendeten Wert an.

RSP_TOUT= 2	Maximale Wartezeit auf die Bestätigung der AT-Kommandos (1 bis 15 Sek.)
# Wahlparameter / Verbindungsaufbau	
DIAL_NUMBER= ?????????	Telefonnummer der Zentrale
DIAL_RETRY= 3	Wahlwiederholung (1 bis 7)
DIAL_TM_REDIAL= 30	Zeit[Sek]: Pause bei Wahlwiederholung (1 bis 255)
DIAL_TM_RIDLE= 1	Zeit[Min]: Sperrzeit nach Wahlwiederholung (1 bis 255)
DIAL_TM_CON= 60	Zeit[Sek]:Wartezeit auf Verbindung "Connect" (1 bis 255)
DIAL_TM_PW= 2	Zeit[Sek]:Wartezeit bis Passwort nach Verbindungsaufbau gesendet wird
DIAL_PRI_PW= ???????	Passwort Primärstation (Leitrechner) maximal 16 Zeichen
DIAL_SEK_PW= ???????	Passwort Sekundärstation (Unterstation) maximal 16 Zeichen

Anmerkung: Die einzelnen Parameter können von dieser Darstellung abweichende Zuweisungen haben.

6.3.3.1 Erläuterung zu den einzelnen Parametern im Standleitungs-Betrieb

PORT

Hier geben Sie die COM-Schnittstelle (1 oder 2) an, die Sie für die Kommunikation mit IEC 60870-5-101 verwenden wollen.

BAUDRATE

Die Baudrate kann zwischen 600 und 9600 Baud eingestellt werden.

PARITY, DATA, STOPBIT

Bei Standleitungsbetrieb oder direkter serieller Kopplung ist die PARITY auf EVEN zu stellen. Die Anzahl Datenbit muss immer 8 sein, die Anzahl Stoppbit immer 1.

TVS2, TNS2, TVM2, TNM2, PAUSE, S2SIGNAL, M2SIGNAL

Über diese Parameter werden die Vor- und Nachlaufzeit für S2 und die Überwachungszeiten für das M2-Signal, sowie die Pausenzeit zwischen den Telegrammen eingestellt. Diese Angaben sind vor allem bei Verwendung von Standleitungsmodems (UEM 202 oder 201) erforderlich. Über die Parameter S2SIGNAL, M2SIGNAL kann das Setzen bzw. Überwachen der jeweiligen Signale abgeschaltet werden.

POLL_TOUT

Innerhalb dieser Überwachungszeit muss die Station angepollt werden, ansonsten wird die Verbindung als gestört gemeldet und kann nur durch ein "Initialisieren der Verbindungsschicht" aktiviert werden.

BUS_TOUT

Die Überwachung **BUS_TOUT** gibt die Überwachungszeit beim Linienbetrieb an, innerhalb der die eigene oder eine andere Station an der Linie (BUS) gepollt werden muss. Ist die Station in diesen Timout gelaufen, erwartet sie ein Initialisieren der Verbindungsschicht.
Bei Punkt-zu-Punkt oder AWD-Betrieb darf der **BUS_TOUT** nicht kleiner als der **POLL_TOUT** eingestellt werden.

CONF_SC

Über diesen Parameter geben Sie vor, wie die Unterstation antworten soll, wenn keine Datentelegramme zur Übertragung anstehen. ON= Empfangsbestätigung mit Einzelzeichen E5, OFF = Kurztelegramm (Telegram fester Länge).

AVS, OKTETT_AVS

AVS ist die Adresse der Verbindungsschicht. Mit dieser wird die Station gepollt. Die Adresse kann 1 oder 2 Oktett (Byte) lang sein. Diese Einstellung muss mit allen Teilnehmern im System über die Kompatibilitätsliste festgelegt werden.

6.3.3.2 Erläuterung zu den einzelnen Parametern im AWD-Betrieb

PARITY, DATA, STOPBIT

Bei den meisten Wähl-Modems oder ISDN-TAs ist die Übertragung von 8 Datenbit nur ohne Paritätsbit möglich. Setzen Sie deshalb den Parameter PARITY= NO. Die Einstellung muss mit der Projektierung auf der Gegenseite übereinstimmen.

RSP_TOUT ist die Überwachungszeit, innerhalb der die Modem-Rückmeldungen (CONNECT, OK etc.) eintreffen müssen. Die vorgegebene Standardeinstellung ist für die meisten Modemtypen passend und muss nicht verändert werden.

Kommt eine von der Unterstation eingeleitete Verbindung nicht innerhalb der mit Parameter **DIAL_TM_CON** angegebenen Zeit zustande, wird der Anwahlversuch abgebrochen. Nach Ablauf der Wartezeit **DIAL_TM_REDIAL** wird erneut versucht die Verbindung aufzubauen. Dies wird solange wiederholt, bis die Verbindung zustande kommt oder die eingestellte Anzahl von Wahlwiederholungen **DIAL_RETRY** erreicht ist.

Ist die vorgegebene Anzahl von Anwahlversuchen erreicht ohne dass eine Verbindung zustande kam, wird ein Timer mit der in **DIAL_TM_RIDLE** angegebenen Zeit gestartet. Nach Ablauf dieser Zeit wird die oben beschriebene Prozedur erneut durchlaufen.

Das Passwort für die Zentrale (**DIAL_PRI_PW)** und die Unterstation (**DIAL_SEK_PW**) kann maximal 16 Zeichen lang sein. Wird das Passwort mit weniger Zeichen angegeben, wird es automatisch mit Leerzeichen aufgefüllt. Groß/Kleinschreibung wird geprüft. Die angegebenen Passwörter müssen mit der Projektierung im Leitsystem übereinstimmen.

Nachdem ein Verbindungsaufbau erfolgreich eingeleitet wurde (Connect vom Modem) sendet der Leitrechner bzw. die Unterstation sein/ihr Passwort. Das Passwort wird immer von der Seite gesendet, die den Verbindungsaufbau eingeleitet hat und von der Gegenseite geprüft. Ist die Prüfung erfolgreich, wird der Pollbetrieb aufgenommen. Wird bei der Passwortprüfung ein Fehler erkannt, unterbricht die prüfende Seite die Verbindung mit ATH.

6.3.4 Parameterdatei für die Datenpunktdefinition bearbeiten (SVC_RTU?)

[RTU_I10X]	
ASDU= 3	ASDU = Stationsadresse
OKT_ASDU= 2	Anzahl Oktett der Adressbyte
OKT_HERK=TRUE	Übertragungsursache mit Herkunftsadresse
OKT_AINF= 3	Anzahl Oktett der Informationsobjektnummer
APDU_LEN=253	Maximale Länge der Blocktelegramme
BLOCK_OBJEKT= TRUE	Blockung von Informationsobjekten erlaubt TRUE / FALSE
BLOCK_ELEMENT= TRUE	Blockung von Informationselementen erlaubt TRUE / FALSE
IEC_CCI_TYP= O	Zählwertabfrage Abfrage Betriebsart [C | D | O]
RTU_COM_STOP=TRUE	Verhalten bei STOO des Anwenderprogramms, TRUE= Kommunikation stoppt, FALSE= läuft weiter
EXEC_CMDNENABLE = TRUE	Befehle mit Qualitätskennung QU=0 sind erlaubt
EXEC_CMDNTIME = 10	Befehlsausführungszeit für Befehle ohne zusätzliche Festlegung (QU=0)
EXEC_STIME= 3	Kurze Befehlsausführungszeit (n * 100 ms)
EXEC_LTIME= 50	Lange Befehlsausführungszeit (n * 100 ms)
1:AP_NMB= 3600	Anzahl Telegramme im Archiv für PVs mit Zeitmarke (0 bis 3600)
1:AP_WARN= 2500	Überlaufwarnung des Archivpuffers (führt bei AWD-Betrieb zum Verbindungsaufbau).
OOFS_SP=100	Objektnummern - Offset Einzelmeldungen (Single Point)
OOFS_SP_TM=100	Objektnummern - Offset Einzelmeldungen (Single Point) mit Zeitmarke
OOFS_DP=200	Objektnummern - Offset für Doppelmeldungen (Double Point)
OOFS_DP_TM=200	Objektnummern - Offset für Doppelmeldungen (Double Point) mit Zeitm.
OOFS_BO=300	Objektnummern - Offset Bitstring
OOFS_BO_TM=300	Objektnummern - Offset Bitstring mit Zeitmarke
OOFS_ME=400	Objektnummern - Offset Messwerte
OOFS_ME_TM=400	Objektnummern - Offset Messwerte mit Zeitmarke
OOFS_IT=500	Objektnummern - Offset Zählwerte
OOFS_IT_TM=500	Objektnummern - Offset Zählwerte mit Zeitmarke
OOFS_SC= 600	Objektnummern – Offset Einzelbefehl (Single Command)
OOFS_DC=700	Objektnummern - Offset Doppelbefehl (Double Command)
OOFS_SE=800	Objektnummern - Offset Sollwert normiert (Set-point command)
OOFS_CBO=900	Objektnummern - Offset Bitstring (Bitstring of 32 Bit))
# interne Anweisung für die Varianten (Datenverarbeitung)	

# Datenvariante Var1	
1:VAR_XX_ABF= TRUE 1:VAR_XX_SPO= TRUE 1:VAR_XX_AWD= FALSE 1:VAR_XX_AP= 1 1:VAR_XX_GRP=1	Abfrage aktiv, Generalabfrage bzw. Zählerabfrage Spontan Übertragung freigeben 1= bei AWD Betreib löst Änderung Verbindungsaufbau aus Zyklisch in Archivpuffer 1 = ja, 0 = nein Wert wird bei GA mit Gruppenabfrage 1 übertragen
# interne Anweisung für die Variante1 nur für Doppelmeldungen(DP)	
1:VAR_DP_TS=100	Störstellungsunterdrückungszeit, 0 = keine Überwachung, n * 10 ms
# interne Anweisung für die Variante1 nur für Bitstring(BO)	
1:VAR_BO_NMB=32	Anzahl Prozesspunkte, 1…32 pro Bitstring
# interne Anweisung für die Variante1 nur für Zählwerte(IT)	
1:VAR_IT_IMPS= 0	Impulsschwelle (threshold) 0 bis 65535 (def. = 0, aus)
# interne Anweisung für die Variante1 nur für Zählwerte (IT) und Messwerte (ME)	
1:VAR_ITME_ZYK = 5	Differenz-/Zykluszeit= n*Minute, 0 bis 3600, 0 = aus
# interne Anweisung für die Variante 1nur für Messwerte (ME)	
1:VAR_ME_AZI=512	Abweichungszeitintegral, AZI = 0… 32760 / 0x7FF8 12 Bit Messwert + VZ, linksbündig (AZI = 8 entspricht der Änderung von einem Digit am Eingang)
1:VAR_ME_OC=32000	Overrange (OV) von 0 bis 32767 (0x7FFF)
# interne Anweisung für die Varianten (Datenverarbeitung) # Datenvariante Var2	
2:VAR_XX_ABF= TRUE 2:VAR_XX_SPONT= TRUE 2:VAR_XX_AP= 0 2:VAR_XX_GRP=2	Abfrage aktiv, Generalabfrage bzw. Zählerabfrage Spontan Zyklisch in Archivpuffer 1 = ja, 0 = nein Wert wird bei GA mit Gruppenabfrage 2 übertragen
# interne Anweisung für die Variante2 nur für Zählwerte(IT)	
2:VAR_IT_IMPS= 20	Impulsschwelle (threshold) 0 bis 65535 (def. = 0, aus)
# interne Anweisung für die Variante1 nur für Zählwerte (IT) und Messwerte (ME)	
2:VAR_ITME_ZYK = 0	Differenz-/Zykluszeit= n*Minute, 0 bis 3600, 0 = aus
# interne Anweisung für die Variante2 nur für Messwerte (ME)	
2:VAR_ME_AZI=256	Abweichungszeitintegral, AZI = 0… 32760 / 0x7FF8

	12 Bit Messwert + VZ, linksbündig (AZI = 8 entspricht der Änderung von einem Digit am Eingang)
2:VAR_ME_OC=32000	Overrange (OV) von 0 bis 32767 (0x7FFF)
#interne Anweisung für die Varianten (Datenverarbeitung) # Datenvariante Var3	
3:VAR_XX_ABF= TRUE 3:VAR_XX_SPO= FALSE 3:VAR_XX_AP= 1	Abfrage aktiv, Generalabfrage bzw. Zählerabfrage Keine spontane Übertragung Zyklisch in Archivpuffer 1 = ja, 0 = nein
# interne Anweisung für die Variante1 nur für Zählwerte (IT) und Messwerte (ME)	
3:VAR_ITME_ZYK = 5	Differenz-/Zykluszeit= n*Minute, 0 bis 3600, 0 = aus
# interne Anweisung für die Variante2 nur für Messwerte (ME)	
3:VAR_ME_AZI=0	Abweichungszeitintegral, AZI = 0… 32760 / 0x7FF8 12 Bit Messwert + VZ, linksbündig (AZI = 8 entspricht der Änderung von einem Digit am Eingang)
	Es können bis zu 16 verschiedene Verarbeitungsvarianten definiert werden.
# Verknüpfung (Zuordnung) von Prozessdaten in Melderichtung (Monitoring Direction)	
QX0.0:TAG_SP_TM= 1;V1 QX0.0:TAG_SP= 1;V1 QX0.1:TAG_SP= 2;V1 QX0.4:TAG_DP=1;V1 QD4:TAG_BO=1;V2 QW8:TAG_ME= 1;V1; QW10:TAG_ME= 2;V2; QW12:TAG_ME_TM= 3;V1; QW14:TAG_ME_TM= 4;V2; QD16:TAG_IT= 1;V1 QD20:TAG_IT_TM= 2;V1	Einzelmeldung mit Zeitstempel Einzelmeldungen ohne Zeitstempel Doppelmeldung Bitstring 32 Bit Messwerte ohne Zeitstempel definiert als WORD Messwerte mit Zeitstempel definiert als WORD 32 Bit Zählwerte definiert als DWORD

QD24:TAG_IT= 3;V2 QD28:TAG_IT_TM= 4;V2	
# Verknüpfung (Zuordnung) von Prozessdaten in Befehlsrichtung (Control Direction)	
IX0.0:TAG_SC= 1;V1 IX0.1:TAG_SC= 2;V1 IW2:TAG_SE=1;V1 IW4:TAG_SE=2;V1	Im 1. Byte des Übergabefels werden zwei Einzelbefehle eingetragen Im 3. und 4. Byte des Übergabefelds wird ein Sollwert eingetragen Im 5. und 6. Byte des Übergabefelds wird ein Sollwert eingetragen

Erläuterung zu den einzelnen Parametern

ASDU, OKT_ASDU:

Die **A**pplication **S**ervice **D**ata **U**nit ist die Stationsadresse. Die Länge ist nur bei IEC 60870-5-101 variabel, bei IEC 60870-5-104 ist sie fest mit 2 vorgegeben.

OKT_HERK:

Gibt an, ob die Übertragungsursache mit oder ohne Herkunftsadresse übertragen wird. Bei IEC 60870-5-104 ist das Oktett für die Herkunftsadresse immer enthalten, wird von der ALU jedoch fest auf NULL gesetzt.

OKT_AINF:

Länge der Adresse des Informationsobjekts. Sie kann 1, 2 oder 3 Byte groß sein. Die Länge ist nur bei IEC 60870-5-101 variabel, bei IEC 60870-5-104 ist sie fest mit 3 vorgegeben.

APDU_LEN= 30

Die maximale Länge der APDU (Protokolldateneinheit der Anwenderschicht = Telegramm) kann vorgegeben werden. Bei IEC-870-5-104 ist die maximale Läge 253 Byte (255 abzüglich Start- und Längenoktett).

BLOCK_OBJEKT

Dieser Schalter gibt an, ob mehrere PVs als geblockte Informationsobjekte übertragen werden dürfen.

Defaulteinstellung = TRUE, da hierdurch eine optimierte Übertragung der PVs insbesondere bei GA möglich ist. Der Schalter sollte nur auf FALSE gestellt werden, wenn das übergeordnete System diese Art von Blocktelegrammen nicht verarbeiten kann.

BLOCK_ELEMENT

Dieser Schalter gibt an, ob mehrere PVs als geblockte Informationselemente übertragen werden dürfen.

Defaulteinstellung = TRUE, da hierdurch eine optimierte Übertragung der PVs insbesondere bei GA möglich ist. Der Schalter sollte nur auf FALSE gestellt werden, wenn das übergeordnete System diese Art von Blocktelegrammen nicht verarbeiten kann.

IEC_CCI_TYP= O

Die Norm sieht für die Übertragung von Zählwerten 4 Verfahren vor. Betriebsart A, B, C und D. Die Betriebsart A kann durch den Parameter VAR_ITME_ZYK bei der Definition der Verarbeitungsvarianten realisiert werden.

Bei den Betriebsarten B, C und D wird ein Zählerabfragebefehl (CI = Counter Interrogation) vom übergeordneten System geschickt. Die ALU 520 unterstützt nur die Betriebsarten C und D, sowie ein OHP spezifisches Verfahren

(O) bei dem die Zählwerte sofort nach einem Umspeicherbefehle (Zählerabfragebefehl mit FRZ=1, RQT=1…5) mit der Ursache 37 … 41 übertragen werden. Dies ist eine Abwandlung der Betriebsart C (eine Zählerabfrage FRZ=0, RQT=1…5 wird hier nicht gesendet).

RTU_COM_STOP

Über diesen Schalter wird vorgegeben, ob die Kommunikation zum Übergeordneten System bei AWP-Stop (AWP = Anwenderprogramm) ebenfalls gestoppt werden soll oder nicht.

FALSE = Kommunikationstreiber laufen weiter und antworten auf Pollaufrufe mit Kurzantwort oder E5. Alle projektierten PVs werden bei AWP-Stopp mit gesetztem Invalid-Bit übertragen. Wird das AWP wieder gestartet werden alle Invalid-Bits zurückgesetzt und die PVs werden übertragen.

RTU_TIME_SET

0 = Zeit darf über IEC gestellt werden (default)
1 = Zeit darf nie über IEC gestellt werden
2 = Zeit darf nur über IEC gestellt werden, wenn aktuelle Zeit IV (ungültig en: invalid) ist

RTU_TIME_ACK_NEG

FALSE = IEC Zeittelegramm wird mit positiver Response bestätigt (default)
TRUE = IEC Zeittelegramm wird mit negativer Response bestätigt

Dieser Parameter ist nur gültig, wenn RTU_TIME_SET ungleich NULL ist!

RTU_TIME_SYNC_ENABLE

Eingabe in Minuten (1…65535)
Delta-Zeitangabe in Minuten, seit dem letzten Stellen der Uhr. Nach Ablauf darf die Uhr über IEC-Telegramm gestellt werden.
Dieser Parameter ist nur gültig, wenn RTU_TIME_SET = 2 ist!

EXEC_CMDNENABLE, EXEC_CMDNTIME:

Wenn der Parameter EXEC_CMDNENABLE = TRUE gesetzt wird, werden Befehle mit der Qualitätskennung OU = 0 (ohne zusätzliche Festlegung) entgegen genommen und mit der in Parameter EXEC_CMDNTIME festgelegten Zeit ausgegeben. Wird der Parameter EXEC_CMDNENABLE = FALSE gesetzt, werden Befehle mit QU=0 negativ bestätigt und nicht ausgeführt.

Normalerweise wird diese Einstellung verwendet, wenn die Befehlsausführungszeit in der Station für jeden Befehl individuell festgelegt wird. Sollte dies im Anwenderprogramm so realisiert sein, ist zu beachten, dass die ALU 504/518 die "Beendigung der Aktivierung" (Ursache 10) immer nach Ablauf der durch EXEC_CMDNTIME vorgegebenen Zeit sendet.

EXEC_STIME, EXEC_LTIME:

Hier wird die kurze und lange Befehlsausgabezeit vorgegeben. Mit welcher Zeit der Befehle ausgegeben wird, wird durch die Befehlskennung im Telegramm von der Zentrale festgelegt. Zeitbasis: n * 100 ms.

1:AP_NMB, 1:AP_WARN:

Es wird die Anzahl von Telegrammen die der Archivpuffer fassen soll vorgegeben (AP_NMB).

Der Parameter 1:AP_WARN ist nur bei AWD-Betrieb aktiv. Er gibt an, bei welchem Füllstand des Puffers von der Unterstation automatisch eine Verbindung zum übergeordneten System eingeleitet wird.

OOFS_??:

Über diese Parameter können Sie pro Datentyp einen Offset vorgeben, der auf die bei der Definition der Eingänge angegebene Objektnummer aufaddiert wird. Die Verwendung der Offsets ermöglicht es **alle** Objektnummern eines Datentyps in einen anderen Bereich zu verschieben.

Im obigen Beispiel werden die ersten 2 Bit im ersten Übergabebyte als Einzelmeldungen mit den Objektnummern 101 und 102 übertragen da der Parameter OOFS_SP mit 100 angegeben ist.

Bit 4 und 5 werden als Doppelmeldung mit der Objektnummer 201 übertragen.

Byte 4..7 des Übergabeblocks (definiert als DWORD) werden als Bitstring mit der Objektnummer 301 übertragen.

Die 4 Messwerte, die ab Byte 8 (definiert als WORD) übergeben werden, werden jeweils mit und ohne Zeitstempel mit den Objektnummern 401, 402, 403 und 404 übertragen.

Die 4 Zählwerte, die ab Byte 16 (definiert als DWORD) übergeben werden mit den Objektnummern 501, 502, 503 und 504 übertragen.

Hinweis: Es wird empfohlen die Offsets für PVs mit und ohne Zeitstempel gleich einzustellen und die Eingänge fortlaufend durchzunummerieren. Dadurch können bei einer Generalabfrage die Daten optimiert übertragen werden.

Verarbeitungsvarianten:

Es können bis zu 8 unterschiedliche Verarbeitungsvarianten (1:VAR…, 2:VAR… usw.) definiert werden. Diese werden anschließend den einzelnen Eingängen zugeordnet. Es gibt Parameter die für alle Datentypen gültig sind (gekennzeichnet mit XX), andere beziehen sich nur auf bestimmte Datentypen (z.B. VAR_IT_IMPS gültig nur bei Zählwerten).

?:VAR_XX_ABF

TRUE = Prozessvariable wird bei Generalabfrage (global) übertragen.

?: VAR_XX_GRP=1

Über diesen Parameter kann die Generalabfrage als Gruppenabfrage (Gruppe 1…16) ausgeführt werden.

?:VAR_XX_SPO

TRUE = Prozessvariable wird bei Änderung spontan übertragen

?:VAR_XX_AWD

TRUE = Einleiten des Verbindungsaufbau zur Zentrale bei Änderung der Prozessvariablen

?:VAR_XX_AP

Prozessvariable zyklisch ins Archiv schreiben. 1 = ja, 0 = nein

?:VAR_BO_NMB

Über diesen Parameter geben Sie an, wie viele hintereinander liegende Bit im Übergabefeld in einem Bitstring zusammengefasst werden. Bereich: 1…32

?:VAR_IT_IMPS

Impulsschwelle im Bereich 0 bis 65535. Bei Erreichen der projektierten Impulsschwelle wird der Zählwert spontan übertragen (Ursache 3). Einstellung 0 bedeutet, dass der Zählwert nur bei Zählerabfrage oder, falls projektiert, zyklisch übertragen wird.

?:VAR_ITME_ZYK

Zeitintervall für die zyklische Übertragung von Zählwerten und Messwerten. Die Werte werden je nach Einstellung von Parameter OFFS_IT / OFF_ME oder OOFS_IT_TM / OOFS_ME_TM mit oder ohne Zeitmarke und der Ursache 3 (spontan) übertragen.

Angabe in Minuten von 1 bis 3600. Einstellung 0 = keine zyklische Übertragung.

?:VAR_ME_AZI

Einstellen des Abweichungszeitintegrals. Die Messwertänderung wird ab dem Zeitpunkt der letzten Übertragung in 1 Sekunden Abständen erfasst und vorzeichenrichtig aufaddiert. Erreicht das Ergebnis dieser Addition den eingestellten AZI, wird der Messwert mit der Ursache 3 (spontan) übertragen. Durch dieses Verfahren wird die Messwertübertragung beruhigt. Langsam ansteigende Messwerte werden verzögert übertragen, schnell ansteigende Werte werden früher übertragen.

Bei Einstellung AZI = 0 wird jede Messwertänderung übertragen falls der Parameter VAR_XX_SPO=TRUE ist. Beachten Sie hierbei, dass Messwerteeingänge generell schwanken.

?: VAR_ME_OV

Hier kann die Grenze für den Überlauf (en:Overrange) projektiert werden. Überschreitet der Messwert den hier angegebenen Wert, wird das Überlaufbit in der Qualitätskennung gesetzt. Einstellung 0 = keine Überwachung.

Messwerte werden 1:1 aus dem Übergabefeld übernommen und ins Telegramm eingetragen. Es ist durch das Anwenderprogramm sicher zu stellen, dass die Messwerte im durch die Norm festgelegten Format übergeben werden.

Definition der Übergabeblöcke:

Syntax: Referenz Übergabefeld:IEC-Typ=Objektnummer;Verarbeitungsvariante

Die Referenz auf das Übergabefeld kann als QX?.? (BIT) QB (BYTE), QW (WORD) oder QD (DWORD) angegeben werden. In Befehlsrichtung werden IB?.?, IB, IW und ID verwendet.

Die Referenz ist **relativ** und bezieht dich immer auf das für die Service-Routine definierte Übergabefeld. Wurde in der I/O Konfiguration von MULTIPROG der Output für den Service z.B. von QB501 bis QB628 definiert, legt QB0 also die Objektdefinition/Verarbeitung der in QB501 eingetragenen Daten fest.

Maximale Länge der Übergabeblöcke:

Befehlsrichtung	512 Byte
Melderichtung	1024 Byte

Meldungen werden über QX?.? definiert. Bei Doppelmeldungen wird immer das erste der beiden aufeinander folgenden Bit als Referenz angegeben.

Messwerte (16 Bit) werden über QW definiert

Bitstrings und Zählwerte (32 Bit) werden über QD definiert.

Befehle werden über IX?.? definiert wobei bei Doppelbefehlen jeweils das erste Bit der beiden aufeinander folgenden Bit als Referenz angegeben wird.

Sollwerte (16 Bit) werden über IW definiert

Bitstrings (32 Bit) werden über ID definiert.

Beispiele

QX0.0:TAG_SP_TE= 1;V1

QX0.0:TAG_SP= 1;V1

Das 1. Bit im Übergabeblock wird als Einzelmeldung (en: **S**ingle **P**oint) sowohl mit, als auch ohne Zeitstempel übertragen. Für die Übertragung gelten die in Variante 1 projektierten Parameter.

QD16:TAG_IT= 1;V1

Der in Byte 16 bis 19 übergebenen Zählwert wird mit den in Verarbeitungsvariante 2 festgelegten Parametern übertragen. Im Beispiel immer nach 5 Minuten und bei Zählerabfrage.

QD28:TAG_IT= 4;V2

Der in Byte 28 bis 31 übergebenen Zählwert wird mit den in Verarbeitungsvariante 2 festgelegten Parametern übertragen. Im Beispiel immer nach 5 Zählimpulsen mit Zeitstempel und bei Zählerabfrage.

TAGs der IEC-Typen:

TAG_SP	Einzelmeldungen (en: **S**ingle **P**oint Information)
TAG_SP_TM	Einzelmeldungen mit Zeitmarke (en: **S**ingle **P**oint Information)
TAG_DP	Doppelmeldungen (en: **D**ouble **P**oint Information)
TAG_DP_TM	Doppelmeldungen mit Zeitmarke (en: **D**ouble **P**oint Information)
TAG_IT	Zählwert (en: **I**ntegrated **T**otal)
TAG_IT_TM	Zählwert mit Zeitmarke (en: **I**ntegrated **T**otal)
TAG_ME	Messwert normiert (**Me**asured Value normalized)
TAG_ME_TM	Messwert normiert mit Zeitmarke (**Me**asured Value normalized)
TAG_BO	Bitmuster von 32 Bit (en: **B**itstring **O**f 32 Bit)
TAG_BO_TM	Bitmuster von 32 Bit mit Zeitmarke (en: **B**itstring **O**f 32 Bit)
TAG_SC	Einzelbefehl (en: **S**ingle **C**ommand)
TAG_DC	Doppelbefehl (en: **D**ouble **C**ommand)
TAG_SE	Sollwert-Stellbefehl normalisiert (en: Set-point command normalized value)
TAG_CBO	Bitmuster von 32 Bit Befehlsrichtung (en: **B**itstring **O**f 32 Bit command direktion)

6.4 Projektierung der Initialisierungsdateien für Modbus RS232-Slave oder TCP-Server

Die Datei MODBUS_SERVER.INI ist im Verzeichnis PCOS_OHP abgelegt und gilt sowohl für den Modbus RS232-Slave wie auch für den Modbus TCP-Server.

☞ **Hinweis:** Ab Firmware-Version 1.70 werden geänderte Dateinamen verwendet (MBSRV.INI)

```
#=====================================
# MODBUS SERVER / SLAVE Konfigurationsdatei
# MODBUS SERVER / SLAVE configuration file
#=====================================

#-----------------------------------------------------
# Interface 1 - SERIAL - Seriell
#-----------------------------------------------------

# Communicationparameter: Modbus Slave Serial Interface - Service 1

[IF1-COM-MODBUS-SSERIAL]
PORT= 2                                 # PORT [xx], 0 ... aus
BAUDRATE= 9600
PARITY= EVEN
DATA=8
STOPBIT=1                               # Stoppbit (1= default)
PAUSE=10                                # mindesten Pausenzeit [ms] zwischen Request u. Response
BUS_TOUT= 2000                          # Busueberwachung (Slave) in n*[ms], def. (2000), Bereich 1000 bis 2000
ASCII= FALSE                            # Protokoll ASCII FALSE/TRUE; (def.FALSE -> RTU)
SLAVE_ADR= 1

# Serverparamter für Modbus Slave Serial interface - Service 1

[IF1-SERV-MODBUS]                               # 0/1 default (0)
REF_BASEINDEX= 0
STATUS_OFFSET= 300
STATUS_LEN= 6                                   # 2, 4 oder 6 Byte lang


# 0x COILS - Read/Write - 1 bit access
REF0_OFFSET= 0                                  # REF0: Offset in Byte-Modbus Server
REF0_LEN=50                                     # Länge in Byte (50 x 8 COILS)

# 1x Discret Input - Read - 1 bit access
REF1_OFFSET= 50                                 # REF1: Offset in Byte-Modbus Server
REF1_LEN=50                                     # Länge in Byte  (50 x 8 Discrete Inputs)

# 3x Input Register - Read - 16 bit access
REF3_OFFSET= 100                                # REF3: Offset in Byte-Modbus Server
REF3_LEN=100                                    # Länge in Byte (100 entsp. 50 Input Register)

# 4x Holding Register - Read/Write - 16 bit access
REF4_OFFSET= 20                                 # REF4: Offset in Byte-Modbus Server
REF4_LEN=100                                    # Länge in Byte (100 entsp. 50 Holding Register)


#-----------------------------------------------------------------------------
# Interface 2 - Modbus TCP/IP Server (MODBUS-STCP_IP)
#-----------------------------------------------------------------------------

# Communicationparameter: Modbus - STCP_IP interface 2


[IF2-COM-MODBUS-STCP_IP]
//IP_ADDR= 192.168.0.101                #IP-Adresse des Teilnehmers
//IP_ADDR= 192.168.0.102                #IP-Adresse des Teilnehmers
```

```
//IP_ADDR= 192.168.0.103          #IP-Adresse des Teilnehmers
//IP_ADDR= 192.168.0.104          #IP-Adresse des Teilnehmers

# Serverparamter für Modbus TCP interface 2

[IF2-SERV-MODBUS]                 # 0/1 default (0)
REF_BASEINDEX= 0
STATUS_OFFSET= 300
STATUS_LEN= 6                     # 2, 4 oder 6 Byte lang


# 0x COILS - Read/Write - 1 bit access
REF0_OFFSET= 0                    # REF0: Offset in Byte-Modbus Server
REF0_LEN=50                       # Länge in Byte (50 x 8 COILS)
# 1x Discret Input - Read - 1 bit access
REF1_OFFSET= 50                   # REF1: Offset in Byte-Modbus Server
REF1_LEN=50                       # Länge in Byte  (50 x 8 Discrete Inputs)
# 3x Input Register - Read - 16 bit access
REF3_OFFSET= 100                  # REF3: Offset in Byte-Modbus Server
REF3_LEN=100                      # Länge in Byte (100 entsp. 50 Input Register)
# 4x Holding Register - Read/Write - 16 bit access
REF4_OFFSET= 20                   # REF4: Offset in Byte-Modbus Server
REF4_LEN=100                      # Länge in Byte (100 entsp. 50 Holding Register)
```

6.4.1 Erläuterung zu den einzelnen Parametern

Mit [IF1…], [IF2…] , [IF3…] usw. wird angegeben auf welches Interface sich die darauf folgenden Parameter beziehen.

[IF1-COM-MODBUS-SSERIAL]

[IF-1-COM-MODBUS-SSERIAL] gibt an, dass diese Einstellungen für das Interface1 mit einer seriellen Modbus-Kopplung gelten.

PORT

Hier geben Sie die COM-Schnittstelle (1 oder 2) an, die Sie für die Kommunikation mit Modbus RS232-Slave verwenden wollen.

BAUDRATE

Die Baudrate kann zwischen 600 und 9600 Baud eingestellt werden.

PARITY, DATA, STOPBIT

Bei der Parität kann zwischen EVEN / ODD bzw. GERADE / UNGERADE gewählt werden. Die Anzahl der Datenbit muss immer 8 sein, die Anzahl Stoppbit immer 1.

PAUSE

Pausenzeit in ms zwischen Anfrage und Antwort. Über diesen Parameter kann das Antworttelegramm verzögert werden, falls der Master nicht in der Lage ist, eine Antwort sofort entgegen zu nehmen.

BUS_TOUT

Busüberwachung in ms. Bereich 1000 bis 2000, Defaulteinstellung 2000 (2 Sekunden).

Der Slave prüft, ob innerhalb der eingestellten Überwachungszeit auf dem Bus Telegramme übertragen werden. Ist dies nicht der Fall, wird der Status "Kommunikationsstörung" gesetzt.

ASCII

Schaler für Modbus-ASCII oder Modbus-RTU. Derzeit ist **nur** Modbus-RTU möglich, Einstellung ASCII=FALSE.

SLAVE_ADR
Hier stellen Sie die Slave- bzw. Node- Adresse ein.

[IF1-SERV-MODBUS]

Die folgenden Anweisungen und Parameter gelten für das Interface1 und definieren den Bereich, der über Modbus gelesen und geschrieben werden kann.

REF_BASEINDEX= 0

Hier stellen Sie ein, ob der Modbus-Master das 1. Coil bzw. Register mit 0 oder 1 adressiert.
Defaulteinstellung ist 0.

STATUS_OFFSET, STATUS_LEN

Über diese beiden Parameter wird festgelegt, ab welcher MB3.???? Adresse der Modbus Status abgelegt wird und wie viele Byte für den Status angezeigt werden sollen. Die Länge kann mit 2, 4 oder 6 Byte vorgegeben werden.

Weiterführende Informationen siehe Kapitel 5.7.4.

Der für die Modbus Kommunikation vorgesehene Shared Memory Bereich kann aufgeteilt werden in Bit- und Wort-Bereiche. Hierfür kann jeweils ein Offset und die Länge des Bereichs angegeben werden. Diese Bereiche können dann mit den entsprechenden Funktionscodes gelesen bzw. beschrieben werden.

In Anlehnung an andere Modbus Projektierungen wurden diese Bereiche wie folgt definiert:

0er Referenz = Ausgangs- oder Merkerbits
1er Referenzen = Eingangsbit
3er Referenzen = Eingangswort
4er Referenzen = Ausgangs- oder Merkerwort

REF0_OFFSET, REF0_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 1	Lese Bit	Read Coils
FC 5	Schreibe einzelnes Bit	Write Single Coil
FC 15	Schreibe Bitbereich	Write Multiple Coils

REF1_OFFSET, REF1_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 2	Lese Eingang	Read Discrete Inputs

REF3_OFFSET, REF3_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 4	Lese Eingangswort	Read Input Register

REF4_OFFSET, REF4_LEN

Dieser Bereich kann mit den folgenden Funktionscodes verarbeitet werden

FC 3	Lese Register	Read Holding Register
FC 6	Schreibe einzelnes Register	Write Single Register
FC 16	Schreibe Registerbereich	Write Multiple Register
FC 23	Lese/Schreibe Registerbereich	Read/Write Multiple Register

Die Bereiche können auch überlappend projektiert werden.

Mit der Einstellung

REF0_OFFSET= 0
REF0_LEN=50

REF4_OFFSET= 0
REF4_LEN=50

können die ersten 50 Byte des Shared Memory sowohl mit den Funktionscodes FC1, FC5 und FC15 wie auch mit den Funktionscodes FC3, FC6, FC16 und FC23 gelesen bzw. geschrieben werden.

[IF2- COM-MODBUS-STCP_IP]

[IF-2-COM-MODBUS-STCP_IP] gibt an, dass diese Einstellungen für das Interface2 mit einer Modbus TCP-Server Kopplung gelten.

IP_ADDR= ?????
Falls die Modbus TCP-Kopplung nur für bestimmte Clients freigegeben werden soll, können deren IP-Adressen angegeben werden. Es können bis zu 4 Verbindungen gleichzeitig aufgebaut werden. Wird keine IP-Adresse angegeben, erhalten die ersten 4 Clients zugriff.

[IF2-SERV-MODBUS]

Die folgenden Anweisungen und Parameter gelten für das Interface2 und definieren den Bereich, der über Modbus gelesen und geschrieben werden kann.

Beschreibung für Baseindex, Status und Referenzen siehe oben.

Mit der Beispielprojektierung kann der gleiche Speicherbereich mit einer seriellen Modbus wie auch mit einer Modbus TCP/IP Kopplung abgefragt und beschrieben werden, wenn beide Interfaces in der Datei PCOS_A520.INI freigegeben werden.

Den beiden Kopplungen können jedoch auch getrennte Bereiche zugewiesen werden. Dies ist dann entsprechen im Shared-Memory und bei der Definition der Referenzen zu berücksichtigen.

7 Firmwarebibliotheken

7.1 Allgemeines

Für die Multiprog-Version 4.6 wurde die Bibliotheken ALUX2X_OHP_??? entwickelt. Mit der Umstellung auf Multiprog 4.8 und der Integration der Funktion "Änderungen senden" (Online-Ändern) wurden diese Firmwarebibliotheken überarbeitet. Diese neuen Bibliothek haben den Namen ALUX2X_OHP4_???.

Um für ältere Projekte kompatibel zu bleiben, werden bei der Installation der OHP-Add-Ons beide Varianten installiert. Die neuen Projekt-Vorlagen beinhalten jedoch nur die neue Varianten ALUX2X_OHP4_???.

Wenn Sie mit einem älteren Projekt die Leistung "Änderungen senden" nutzen möchten, müssen Sie die Bibliotheken austauschen. Klicken Sie hierzu im Projektbaum-Menü den Eintrag *Bibliotheken* mit der rechten Maustaste an. Danach können Sie über *Einfügen – Firmware-Bibliothek* die neuen Bibliotheken einbinden.

☞ **Pfad zu den FW-Bibliotheken:** [Lw]:\Programme\OHP\Multiprog_48\Multiprog\PLC\FW_LIB\

Die alten Bibliotheken können durch Anwahl mit der Maus und Betätigen der Taste *Entf* gelöscht werden.

Voraussetzung für das Online-Ändern ist, dass im Projekt bereits die neuen Ressourcen verwendet werden, und die ALU 5xx eine Firmware ab Version 3.00 enthält.

🕮 Weiterführende Informationen hierzu siehe **Teil II Projektierung und Programmierung,** Kapitel 8.1.

Nach Austausch der Bibliotheken muss das Projekt einmal komplett neu übersetzt (*Code – Projekt neu erzeugen*) und gesendet werden Anschließend steht die Funktion "Änderungen senden" zur Verfügung.

OHP stellt folgende Bibliotheken zur Verfügung:

Für PROCONOS 3

ALUX20_OHP_SCOM

ALUX20_OHP_SEAB

ALUX20_OHP_SYS

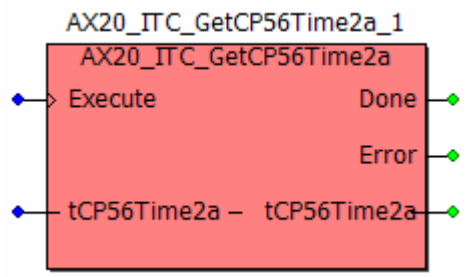
Für PROCONOS 4 (Online-Ändern)

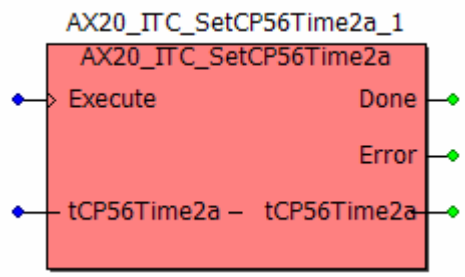
ALUX20_OHP4_SCOM	(Funktionen zu freien Kommunikation über die RS232-Schnittstelle)
ALUX20_OHP4_SEAB	(Funktionen für die Kopplung über IKOS)
ALUX20_OHP4_SYS	(Diverse allgemeine Funktionen)
ALUX20_OHP4_SYS_SA	(Funktionen zur Störungsanalyse und zur Auswertung für DEA – Ankopplung)
ALU352 OHP4 AWPWDG	(Bausteinen zur Aktivierung eines Hardware-Watchdogs über die Anwendertask)

7.2 Bibliothek ALUX20_OHP4_SYS

7.2.1 Auslesen und Stellen der ALU Systemzeit im Anwenderprogramm

Funktionsbaustein AX20_ITC_GetCP56Time2a und AX20_ITC_SetCP56Time2a





Eingangsvariable (VAR_IN):
Execute : BOOL Freigabeflag für den Funktionsbaustein, 1 = ausführen

Ausgangsvariable (VAR_OUT):
Done : BOOL 1 = Funktion ausgeführt
Error : BOOL 1 = Funktion fehlerhaft

Ein-/Ausgangsvariable (VAR_IN_OUT):
tCP56Time2a : 7 Byte Systemzeit (Datum/Uhrzeit)

Diese Systemzeit wird nach IEC 60870-5-4, CP56Time2a in 7 Byte wie folgt dargestellt:

<table>
<tr><td>2^7</td><td colspan="4">Millisekunden</td><td>2^0</td></tr>
<tr><td>2^{15}</td><td colspan="4">0…59999</td><td>2^8</td></tr>
<tr><td>IV</td><td>Res</td><td>2^5</td><td colspan="2">Minute 0…59</td><td>2^0</td></tr>
<tr><td>S</td><td colspan="2">Res</td><td>2^4</td><td>Stunden 0…23</td><td>2^0</td></tr>
<tr><td>2^7</td><td>WT 1..7</td><td>2^5</td><td>2^4</td><td>Tag im Monat 1…31</td><td>2^0</td></tr>
<tr><td colspan="2">Res</td><td colspan="2">2^3</td><td>Monat 1…12</td><td>2^0</td></tr>
<tr><td>Res</td><td>2^6</td><td colspan="3">Jahre 0…99</td><td>2^0</td></tr>
</table>

IV = Ungültig (en: invalid)

IV <0> = gültig
IV <1> = ungültig

S = Sommerzeit

S<0> = Normalzeit
S<1> = Sommerzeit

WT = Wochentag

WT<0> = keine Angabe
WT<1> = Montag

:
WT<7> = Sonntag

In Multiprog kann diese Zeit z.B. in einem Array oder in einer Struktur abgebildet werden.

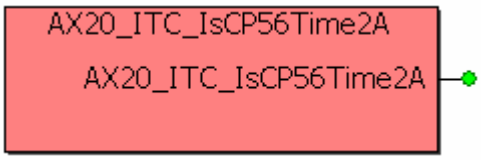
```
TYPE
   ArrayTime      :        ARRAY [1..7] OF BYTE;
END_TYPE


TYPE
IEC_Time:
   STRUCT
         ms:          WORD;
         minute:      BYTE;
         hour:        BYTE;
         day:         BYTE;
         mon:         BYTE;
         year:        BYTE;
   END_STRUCT;
END_TYPE
```

Eine Variable vom Typ IEC_Time oder ArrayTime wird an den Funktionsbaustein AX20_ITC_GetCP56Time2a übergeben um die aktuelle Systemzeit auszulesen, bzw. an den Funktionsbaustein AX20_ITC_SetCP56Time2a um die Zeit zu stellen.

Die OHP-Projekt-Vorlagen beinhalten bereits die Struktur IEC_Time in der Datentyp-Deklaration.

7.2.2 Status der ALU Systemzeit lesen

Funktion AX20_ITC_IsCP56Time2A



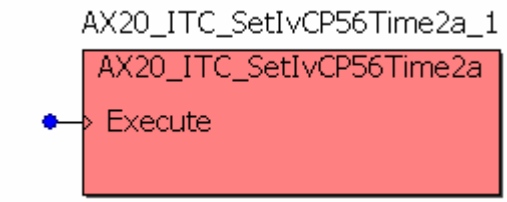
Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

Ausgangsvariable (VAR_OUT):
AX20_ITC_IsCP56Time2A : BOOL Status der Systemzeit

0 = Zeit ist ungültig (invalid)
1 = Zeit ist gültig

7.2.3 Status der ALU Systemzeit auf ungültig setzen

Funktionsbaustein AX20_ITC_SetIvCP56Time2A

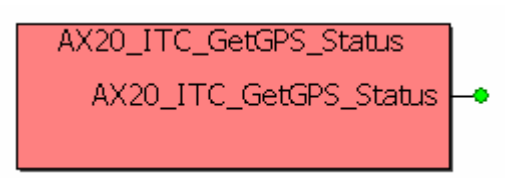


Eingangsvariable (VAR_IN):
Execute : BOOL Freigabeflag für den Funktionsbaustein, 1 = ausführen

Wird dieser Funktionsbaustein ausgeführt, wird die ALU-Systemzeit auf ungültig gesetzt. Dieser Aufruf kann z.B. in einer Systemtask Kalt und/oder Warmstart verwendet werden, um die Systemzeit nach einem Neustart auf ungültig zu setzen, bis sie wieder gezielt gestellt wird (GPS, SNTP).

7.2.4 Status der ALU Systemzeit bei GPS-Synchronisation

Funktion AX20_ITC_GetCP56GPS_Status



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

Ausgangsvariable (VAR_OUT):
AX20_ITC_GetCP56GPS_Status : DWORD Status der Systemzeit bei GPS-Synchronisation

Status in Hex:

16#00000001 GPS-Modul bereit

16#00000003 Uhrzeit gestellt

16#81000001 Uhrzeit kann wegen GPS-Warnung "Uhrzeit ungültig" nicht übernommen werden

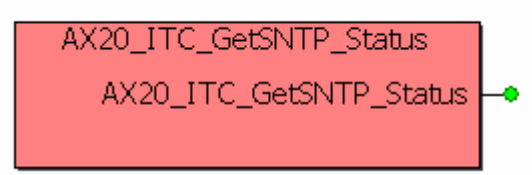
16#82000001 Fehler beim Setzen der Uhrzeit

16#84000001 RMC Datensatzfehler

16#88000001 Keine Verbindung zum GPS-Modul

7.2.5 Status der ALU Systemzeit bei SNTP-Synchronisation

Funktion AX20_ITC_GetCP56SNTP_Status



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

Ausgangsvariable (VAR_OUT):
AX20_ITC_GetCP56SNTP_Status : DWORD Status der Systemzeit bei SNTP-Synchronisation

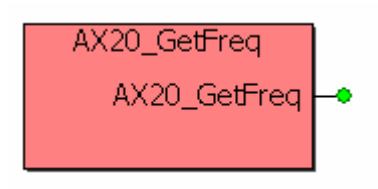
Status in Hex:

16#00000001 Verbindung zum SNTP-Server bereit

16#00000003 Uhrzeit gestellt

7.2.6 Systemtakte

Funktion AX20_ITC_GetFreq



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

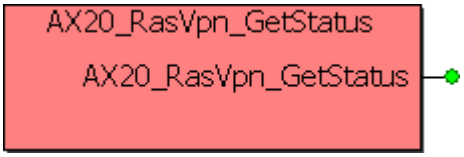
Ausgangsvariable (VAR_OUT):
AX20_ITC_GetFreq : WORD Blinktakte

☞ **Hinweis:** Für die Programmiersprache AWL (en: Instruction List) gibt es einen gesonderten Baustein ILAX20_GetFreq. Die Funktionalität ist die gleiche, jedoch ist der Baustein mit einem zusätzlichen Execute-Eingang versehen.

Bit	Frequenz	Zeitraster
X0	20 Hz	50 ms
X1	10 Hz	100 ms
X2	5 Hz	200 ms
X3	2,5 Hz	400 ms
X4	1,25 Hz	800 ms
X5		nicht definiert
X6		nicht definiert
X7		nicht definiert
X8	2 Hz	500 ms
X9	1 Hz	1 s
X10	0,5 Hz	2 s
X11	0,25 Hz	4 s
X12	0,125 Hz	8s
X13		nicht definiert
X14		nicht definiert
X15		nicht definiert

7.2.7 Status der GPRS-Verbindung

Funktion AX20_RasVpn_GetStatus



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

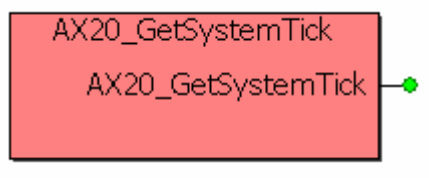
Ausgangsvariable (VAR_OUT):
AX20_RasVpn_GetStatus : DWORD Status der Verbindung

Die Funktion liefert den Status der RAS- und VPN-Verbindung bei Datentransfer über das interne GPRS-Modem.

Status in Hex	Bedeutung
16#00000001	Initialisierung läuft
16#00000002	GPRS-Modul bereit
16#00000003	GPRS-Modul bereit - keine Verbindung
16#00000004	GPRS-Modul bereit - Modem Verbindung abgebrochen
16#00000005	GPRS-Modul bereit - VPN Verbindung/Login abgebrochen
16#00000006	GPRS-Modul bereit - VPN-Verbindung kein Login - PPP abgebrochen
16#00000008	Start Modem-Verbindung
16#00000009	Modem Verbindung ok
16#0000000A	Start VPN-Verbindung
16#0000000B	VPN-Tunnel steht
16#0000000C	Zwangstrennung LOGOUT/LOGIN
16#0000000D	Zwangstrennung nach Trafic
16#0000000E	Hardware-Reset ausgelöst
16#0000000F	Software-Reset ausgelöst
16#00000010	Anzahl Verbindungsversuche zum Provider erreicht
16#00000011	Anzahl Verbindungsversuche VPN erreicht

7.2.8 SystemTick

Funktion AX20_GetSystemTick



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

Ausgangsvariable (VAR_OUT):
AX20_GetSystemTick : DWORD Zeit in Millisekunden

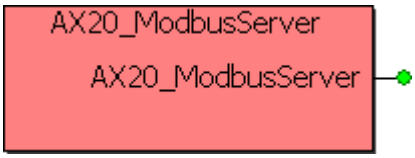
Die Funktion liefert die aktuelle Zeit in Millisekunden seit dem Einschalten der ALU.

Vorwärtszähler mit Umschlagpunkt. Der Wert wird somit alle 49.71 Tage (2^{32} Millisekunden) wieder auf Null gesetzt.

☞ **Hinweis:** Für die Programmiersprache AWL (en: Instruction List) gibt es einen gesonderten Baustein ILAX20_GetSystemTick. Die Funktionalität ist die gleiche, jedoch ist der Baustein mit einem zusätzlichen Execute-Eingang versehen.

7.2.9 Modbus-Server

Funktion AX20_ModbusServer



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionen

Ausgangsvariable (VAR_OUT):
AX20_ModbusServer : INT Status

Status = 0 = Modbus-Treiber läuft nicht
Status = 1 = Modbus-Treiber läuft

Wenn bei der ALU das Übertragungsprotokoll Modbus-RS232-Slave oder Modbus-TCP-Server benutzt wird, sollte dieser Baustein mindestens einmal in der Default-Task am Anfang oder Ende aufgerufen werden. Der Baustein beantwortet die vom Master/Client gesendeten Abfragen. Dadurch wird sichergestellt, dass die Inhalte der einzelnen Register eines Telegramms aus dem gleichen Programmzyklus stammen. Dies ist insbesondere dann wichtig, wenn Werte berechnet werden und in einem bestimmten Verhältnis zueinander stehen.

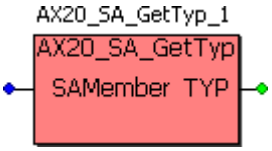
Wird der Baustein nicht verwendet, kann es passieren, dass Inhalte eines Telegramms aus verschiedenen Programmzyklen stammen, da der Modbus-Treiber asynchron zum Programmzyklus arbeitet.

☞ **Hinweis:** Für die Programmiersprache AWL (en: Instruction List) gibt es einen gesonderten Baustein ILAX20_ModbusServer. Die Funktionalität ist die gleiche, jedoch ist der Baustein mit einem zusätzlichen Execute-Eingang versehen.

7.3 Bibliothek ALUX20_OHP4_SYS_SA

7.3.1 Baugruppentyp auslesen bei DEA-Kopplung

Funktionsbaustein AX20_SA_GetTyp



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionsbausteine

Eingangsvariable (VAR_IN):
SAMember : INT Steckplatzadresse/Teilnehmernummer

Ausgangsvariable (VAR_OUT):
Typ : WORD Baugruppentyp

Der Baustein liefert für die Erweiterungsbaugruppenträger die E/A-Teilnehmer der **projektierten** Bestückungsliste zurück.

Der Rückgabewert vom Datentyp WORD (2 Oktett) unterteilt sich in ein Byte für den E/A-Typ (HIGH-Byte) und ein Byte für den Identcode (LOW-Byte)

Definition HIGH-Byte (Typ)

Wert	Bedeutung
16#00	Baugruppe/Teilnehmer nicht projektiert bzw. nicht vorhanden.
16#11	Digital Input z.B. DEP112
16#12	Digital Output z.B. DAP112
16#13	Digital Input und Output (gemischte Karte) z.B. DAP102
16#21	Analog Input wird zurzeit nicht unterstützt!
16#22	Analog Output wird zurzeit nicht unterstützt!
16#80	Experte allgemein
16#83	Experte Kommunikationsbaugruppe
16#FF	Baugruppe im Grundbaugruppenträger (Low-Byte: ID-CODE)

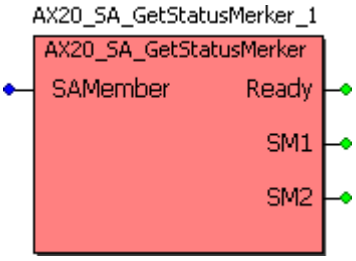
☞ **Hinweis:** Falls das HIGH-Byte 0 ist und im LOW-Byte ein Wert ungleich 0 steht, handelt es sich bei dem betreffenden Teilnehmer um eine DEA-Baugruppe, das LOW-Byte repräsentiert dann die Bitbusadresse der DEA!

Definition LOW-Byte (Identcode)

Wert	Bedeutung
16#01	DEP512/DEP112, 32 Bit Eingabe
16#02	DAP512/DAP112, 32 Bit Ausgabe
16#03	ADU516/ADU116, 16 x Analog Eingänge
16#04	DAU108, 8 x Analogausgänge
16#05	DAP502/DAP102, 16 Bit Eingänge / 16 Bit Ausgänge
16#07	DAP503/DAP103, 16 Bit Eingänge / 16 Bit Ausgänge
16#16	DAP506/DAP106, 16 Bit Ausgabe
16#21	ADU515/ADU115, 16 x Analog Eingänge
16#25	DAU104, 8 x Analogeingänge u. 4 x Analogausgänge
16#34	DAU509/DAU109, 8 x Analogausgänge
16#45	BIK514/BIK114
16#51	Experte ZAE105 i. V.
16#D1	Experte ADEAS ab ALU Firmware-Version 1.89

7.3.2 Baugruppen-Statusmerker

Funktionsbaustein AX20_SA_GetStatusMerker



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionsbausteine

Eingangsvariable (VAR_IN):
SAMember : INT Steckplatzadresse/Teilnehmernummer

Ausgangsvariable (VAR_OUT):
Ready : BOOL 0 = Fehler, 1 = Funktion korrekt ausgeführt
SM1 : BOOL Statusmerker 1
SM2 : BOOL Statusmerker 2

1. Statusmerker

Der Merker wird auf TRUE gesetzt, wenn im ersten Statusmerkerwort eines der Bits 0...5, Bit 7 oder Bit 8 gesetzt ist. In diesem Fall wird im System der E/A-Betrieb mit der fehlerhaften Baugruppe eingestellt. Der Fehler kann durch Normierung (Ab- und Zuschalten der Versorgungsspannung) oder Austausch der Baugruppe behoben werden.

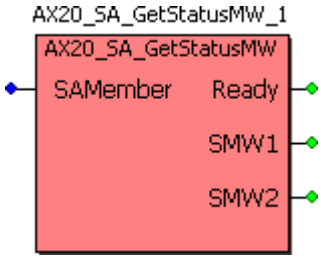
2. Statusmerker

Der Merker ist bei Digitalen Baugruppen immer FALSE. Der Merker wird nur bei Analogen- oder Expertenbaugruppen benutzt. Er wird auf TRUE gesetzt, wenn auf der Baugruppe eine Teilstörung vorliegt.

Diese Störung kann sein: Kanal der analogen Baugruppe fehlerhaft (Bit 13 im ersten Statusmerkerwort) oder Experte meldet einen Sammelfehler (Bit 15 im ersten Statusmerkerwort).

7.3.3 Baugruppen-Statuswort

Funktionsbaustein AX20_SA_GetStatusMW



Eingangsvariable (VAR_IN):
SAMember : INT Steckplatzadresse/Teilnehmernummer

Ausgangsvariable (VAR_OUT):
Ready : BOOL 0 = Fehler, 1 = Funktion korrekt ausgeführt
SMW1 : WORD Statusmerkerwort 1
SMW2 : WORD Statusmerkerwort 2

Der Baustein liefert für die Erweiterungsbaugruppenträger die Statusinformationen der E/A-Teilnehmer zurück. Die Belegung des 1. Statusmerkerworts ist für alle Baugruppen gleich definiert. Das 2. Statusmerkerwort kann durch die Baugruppen individuell genutzt werden.

Definition 1. Statusmerkerwort

Bit	Bedeutung
X0	Baugruppe ausgefallen oder nicht gesteckt
X1	Identcode der Baugruppe entspricht nicht dem Eintrag in der Bestückung
X2	Baugruppe nicht bereit
X3	Hardware-Fehler am PAB
X4	Bitbusprozedur-Fehler (Initialisierung fehlerhaft)
X5	Bitbusverbindung gestört (DEA-Ausfall oder Verbindung unterbrochen)
X6	E/A-Operation wegen Unterspannungs-Warnung abgebrochen
X7	Zugriff mit falscher Teilnehmernummer während Init-Hochlauf
X8	E/A-Operation wegen Baugruppen-Ausfall gesperrt oder DEA-Verbindung gestört

X9	Initialisierung aktiv. Experten-Hochlauf oder DEA-Ankopplung
X10	Baugruppe wegen noch laufender Operation belegt oder Batterie-Ausfall im DEA-Magazin
X11	Unterbrechung der laufenden E/A-Operation wegen Direkt-E/A oder Ablauf der Telegrammüberwachungszeit im DEA-Magazin
X12	Baugruppen-Sammelfahler, das Sammelfehlerbit im Identcode ist gesetzt
X13	Kanalfehler, eine oder mehrere Messstellen der Analog-Baugruppe sind ausgefallen
X14	
X15	Experten Sammelfehler

Definition 2. Statusmerkerwort

Siehe hierzu auch die Beschreibungen der einzelnen Baugruppen.

Analog-Baugruppen: Bei analogen Baugruppen ist in diesem Statuswort, bei Störung eines oder mehrere Messkanäle (Bit 13 im ersten Statuswort), der jeweiligen Kanalnummer (max. 16 Kanäle möglich) ein Fehlerbit zugeordnet.

Experten-Baugruppen: Für Experten-Baugruppen ist, bei Experten-Sammelfehlern (Bit 15 im ersten Statuswort), die Fehlernummer wie folgt belegt:

LOW-Byte (Bit 0...7) enthält zusätzliche Fehlerparameter zur Erläuterung der Fehlernummer.

HIGH-Byte (Bit 8...15) Bit 8 ... 13 = Fehlernummer, Bit 14 = Kennung für gehende Fehler, Bit 15 = Kennung für kommende Fehler

Fehlernummer:

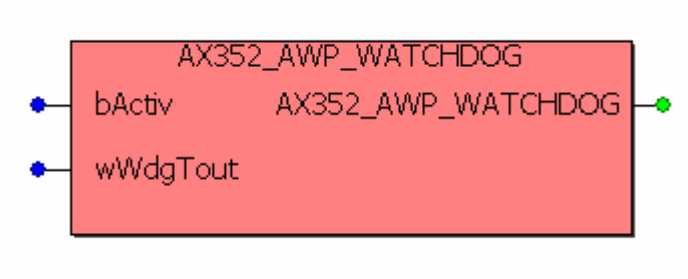
0...15	Synchronisationsfehler, n=SyncVar
16	Teilnehmer mit E/A-Operation belegt
17	letzter Ausgabe-Interrupt läuft noch
18	DPM-Schreibsperre wegen V24-ExpPuTE
19	Sammelfehler bei E/A über PAB
20	Sammelfehler bei E/A über Modnet 1/SFB
21	keine E/A bei Spannungswiederkehr
22	lfd. Ausgabe mit Interrupt abgebrochen
23	Weiterhin Fehler nach Behebung der Verbindungsstörung
24	Ausgabe-(Ziel)-Adresse ist noch belegt
25	Eingabe-(Quell)-Adresse wurde noch nicht angesprochen

7.4 Bibliothek ALU352 OHP4 AWPWDG

Die Bibliothek steht ab der PCOS Version 3.60 zur Verfügung

7.4.1 Hardware-Watchdog über Anwenderprogramm

Funktionsbaustein AX352_AWP_WATCHDOG



Achtung: Verwenden Sie keine EN/ENO für die OHP-Funktionsbausteine

Eingangsvariable (VAR_IN):

bActive	: BOOL	Aktivierung der Funktion
wWdgTout	: WORD	wWdgTout-Zeit kann 0 oder zwischen 100 und 2000 Millisekunden sein

Ausgangsvariable (VAR_OUT):

AX352_AWP_WATCHDOG	: WORD	Statusinformationen

Status in Hex	Bedeutung
16#0000	Anwender-Task-HW-Watchdog aktiv
16#8000	bActiv= FALSE
16#8001	Hardware Watchdog nicht initialisiert, interne Fehler
16#8002	wWdgTout Zeit ungültig

Der Watchdog wird über bActiv = TRUE und die wWdgTout [ms] aktiviert.

Die wWdgTout-Zeit kann maximal 100 bis 2000 Millisekunden sein.

Der Watchdog kann über bActiv = FALSE und wWdgTout-Zeit= 0ms deaktiviert werden.

Zusätzlich werden im Systemmerker-Bereich noch die folgenden Informationen angezeigt:
Systemmerker %MD1.2928: HW Watchdog Status

Wet in Hex	Bedeutung
16#0000	HW Watchdog nicht aktiv
16#0001	HW Watchdog aktiv
16#0002	HW Watchdog inaktiv – Breakpoint gesetzt
16#0004	HW Watchdog inaktiv – Download Changes, etc.

☞ **Hinweis:** Der Hardware-Watchdog ist mit dem Software Watchdog einer ProConOS Anwender Task nicht zu verwechseln. Der HW-Watchdog führt nach dem Auslösen zum Reset der SPS.

Die Funktion ALU352_OHP4_AWPWDG aktiviert einen Hardware-Watchdog zur Überwachung einer Anwendertask (z.B. DEFAULT Task) auf der ALU. Der Watchdog wird über bActiv = TRUE und die wWdgTout [ms] aktiviert. Die wWdgTout-Zeit kann zwischen 100 bis 2000 Millisekunden sein.

Wenn der Watchdog einmal aktiviert wurde, muss die Watchdog-Funktion zyklisch (min. 1 x pro Zyklus z.B. in der DEFAULT Task) in kürzeren Abständen aufgerufen werden als in der durch wWdgTout vorgegebenen Zeit. Die wWdgTout-Zeit sollte mindestens das dreifache der aktuellen Zykluszeit betragen, da bei Ablauf der wWdgTout-Zeit die ALU automatisch einen Hardare-Reset durchführt.

Der Watchdog dient dazu das System automatisch neu zu Booten, wenn es z.B. in eine Endlosschleife gelaufen ist oder wenn die SPS steht.

Der Watchdog kann über **bActiv= TRUE und wWdgTout-Zeit= 0ms** deaktiviert werden.

Bei aktiviertem Watchdog führt der Wechsel von bActive =TRUE auf FALSE (wWdgTout-Zeit wird nicht auf 0 gesetzt) ebenfalls zum Hardware-Reset der ALU. Dies kann genutzt werden, um die ALU Ereignisgesteuert zu Booten.

Nach Ablauf der Watchdog Zeit (wWdgTout) versucht der SPS-Prozess-Scheduler die Anwender TASKs zu beenden, d.h. die SPS geht in den STOP Modus die RUN-LED erlischt und die 1. rote LED blinkt. Anschließend führt das System eine START (WARM/KALT) mit HW-Reset des SPS-Systems durch. Sollte das Stoppen der Anwender TASKs nicht erfolgreich sein bzw. der SPS-Prozess-Scheduler kein Zugriff auf das Betriebssystem haben, erfolgt unmittelbar ein Reboot.

Der Anwender braucht bei DEBUG, Online Trace sowie Download Betriebsarten keine Rücksicht auf die eingestellte HW-Watchdog Zeit zu nehmen. Das SPS Laufzeitsystem erkennt selbstständig die Betriebsart und aktualisiert den Watchdog im Hintergrundbetrieb. Sobald der Anwender einen Breakpoint in seinem Anwenderprogramm setzt wird die RUN LED auf den Zustand Blinken gesetzt. Außerdem wird dies über den Systemmerker %MD1.2928 angezeigt.

Um den HW-Watchdog zu testen wird in einem AWP Zyklus der FWB ALU352_OHP4_AWPWDG mit bActiv = TRUE und wWdgTout= 1000 aufgerufen und im nächsten Zyklus wird bActiv= FALSE gesetzt.

8 Tipps und Tricks

8.1 Update eines bestehenden Multiprog Projekts für Online-Ändern

Voraussetzung für das Online-Ändern ist Multiprog 4.8 und eine ALU 5xx Firmware ab Version 3.00. Die aktuelle Version kann jeweils von der OHP-Webseite geladen werden.

Wenn Sie mit einem älteren Projekt die Funktion Online-Ändern nutzen möchten, müssen Sie folgende Einstellungen prüfen und gegebenenfalls ändern.

Hardwarestruktur:

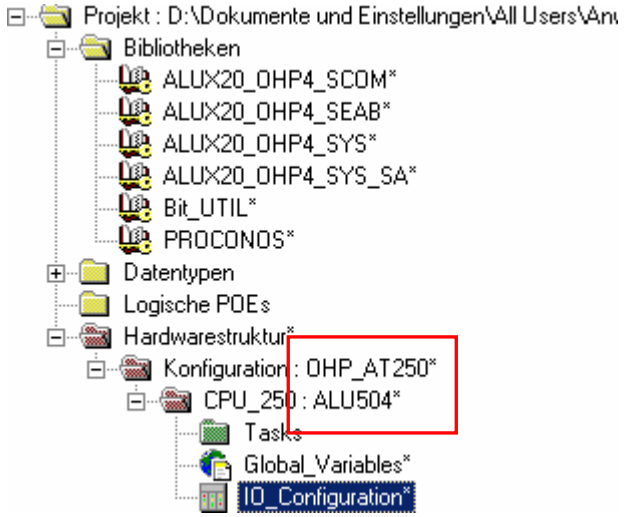
Wenn der SPS-Typ "IPC_33" eingestellt ist, wählen Sie den Eintrag mit der rechten Maustaste an und öffnen Sie den Dialog *Eigenschaften*. Im Register *SPS/Prozessor* wählen Sie dann den *SPS-Typ* "OHP_AT250" aus.

Jetzt muss noch der CPU-Typ eingestellt werden. Hierfür wählen Sie wieder über die rechte Maustaste den Dialog *Eigenschaften* aus und stellen dann im Register *SPS/Prozessor* die entsprechende ALU-Variante ein.

Bibliotheken:

Wenn Sie in ihrem Projekt Funktionen aus den OHP-Firmware-Bibliotheken genutzt haben, müssen Sie diese ebenfalls updaten. Wie Sie hierbei vorgehen ist im **Teil II Projektierung und Programmierung,** Kapitel 7 beschrieben.

Der Projektbaum sollte anschließend wie folgt aussehen.



☞ **Hinweis:** Nur die rot umrandeten Einträge sind relevant und müssen neu eingestellt sein. Die Bezeichnungen "Konfiguration" und "CPU_250" können in Ihrem Projekt evtl. anders lauten und müssen nicht geändert werden. Firmware-Bibliotheken müssen nur dann ausgetauscht werden, wenn Sie Funktionen aus diesen benutzt haben.

8.2 Auslesen und Stellen der ALU504/518 Systemzeit im Anwenderprogramm

OHP stellt den Anwendern eine Firmwarebibliothek (ALUX2X_OHP_SYS) zur Verfügung, die u.a. Funktionen zum Stellen bzw. Auslesen der Systemzeit beinhaltet.

Diese Systemzeit wird nach IEC 60870-5-4, CP56Time2a in 7 Byte wie folgt dargestellt:

<table>
<tr><td>2^7</td><td colspan="5">Millisekunden</td><td>2^0</td></tr>
<tr><td>2^{15}</td><td colspan="5">0…59999</td><td>2^8</td></tr>
<tr><td>IV</td><td>Res</td><td>2^5</td><td colspan="3">Minute 0…59</td><td>2^0</td></tr>
<tr><td>S</td><td colspan="2">Res</td><td>2^4</td><td colspan="2">Stunden 0…23</td><td>2^0</td></tr>
<tr><td>2^7</td><td colspan="2">WT 1..7 2^5</td><td>2^4</td><td colspan="2">Tag im Monat 1…31</td><td>2^0</td></tr>
<tr><td colspan="3">Res</td><td>2^3</td><td colspan="2">Monat 1…12</td><td>2^0</td></tr>
<tr><td>Res</td><td colspan="2">2^6</td><td colspan="3">Jahre 0…99</td><td>2^0</td></tr>
</table>

IV = Ungültig (en: invalid)

IV <0> = gültig
IV <1> = ungültig

S = Sommerzeit

S<0> = Normalzeit
S<1> = Sommerzeit

WT = Wochentag

WT<0> = keine Angabe
WT<1> = Montag
WT<7> = Sonntag

In Multiprog kann diese Zeit z.B. in einem Array oder in einer Struktur abgebildet werden.

```
TYPE
   ArrayTime     :     ARRAY [1..7] OF BYTE;
END_TYPE


TYPE
IEC_Time:
   STRUCT
        ms:          WORD;
        minute:      BYTE;
        hour:        BYTE;
        day:         BYTE;
        mon:         BYTE;
        year:        BYTE;
   END_STRUCT;
END_TYPE
```

Eine Variable vom Typ IEC_Time oder ArrayTime wird an den Funktionsbaustein AX20_ITC_GetCP56Time2a übergeben um die aktuelle Systemzeit auszulesen, bzw. an den Funktionsbaustein AX20_ITC_SetCP56Time2a um die Zeit zu stellen.

8.3 Uhrzeit stellen und ALU-Statusinformationen über Telnet

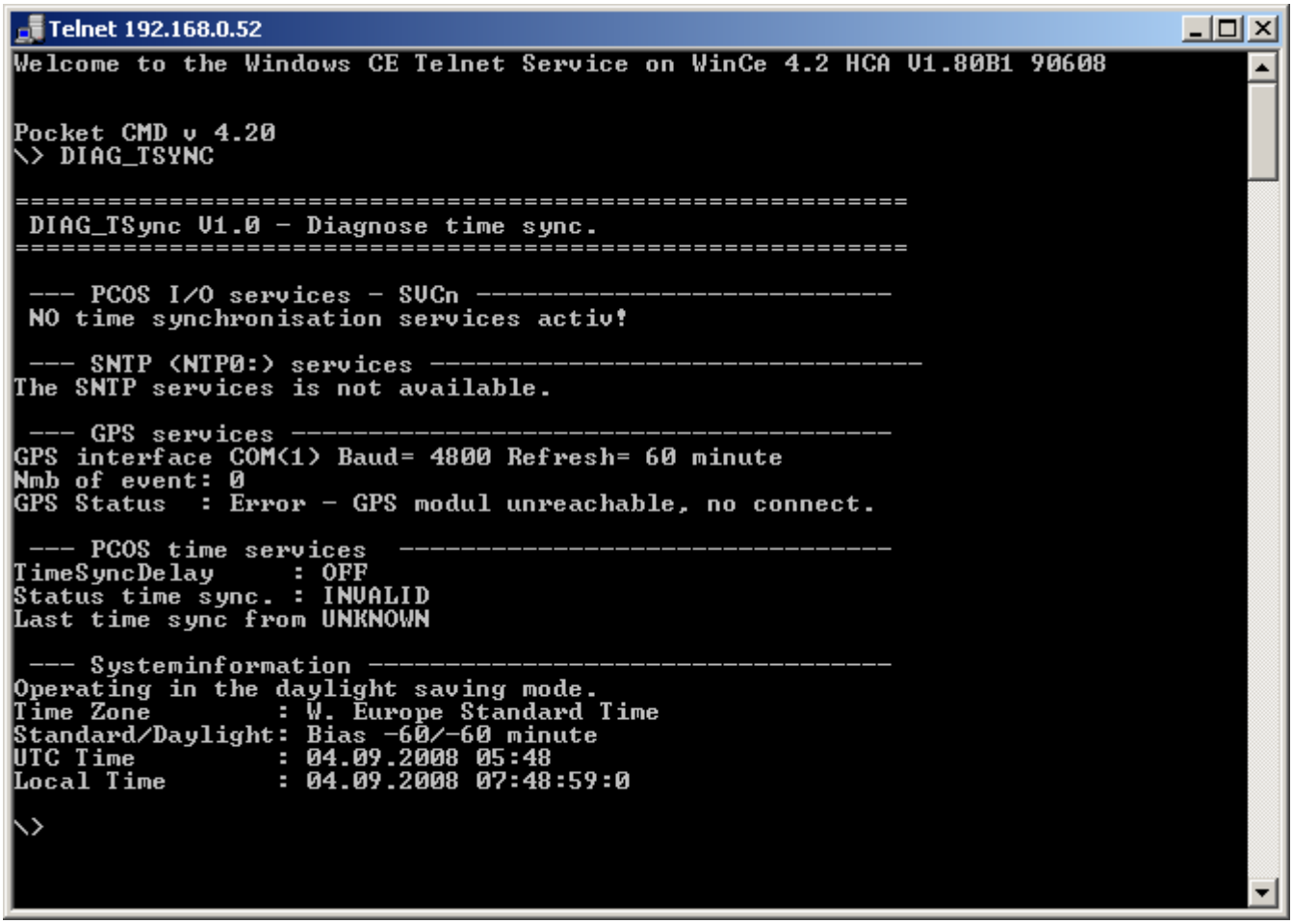
Über eine Telnet-Verbindung können Sie die ALU-Systemzeit stellen oder den Status der ALU-Systemzeit ermitteln.

- Schließen Sie hierfür den Bedien-PC an die Ethernet-Schnittstelle auf der ALU-Unterseite an.
- Geben Sie im Windows Startmenü im Dialog Ausführen *telnet 192.168.0.52* ein (falls Sie die ALU-IP-Adresse geändert haben, geben Sie die geänderte Adresse ein).

In Dialogfenster das anschließend erscheint wird die WindowsCE Version und die Firmware-Version angezeigt.

Es stehen Ihnen jetzt folgende Funktionen zur Verfügung:

- Über *time* wird die aktuelle Uhrzeit angezeigt und kann gestellt werden
- Über *date* wird das aktuelle Datum angezeigt und kann gestellt werden
- *DIAG_TSYNC* zeigt den Status der ALU-Systemzeit an


Telnet 192.168.0.52
Welcome to the Windows CE Telnet Service on WinCe 4.2 HCA V1.80B1 90608
Pocket CMD v 4.20
\> DIAG_TSYNC
DIAG_TSync V1.0 - Diagnose time sync.
--- PCOS I/O services - SVCn
NO time synchronisation services activ!
--- SNTP (NTP0:) services
The SNTP services is not available.
--- GPS services
GPS interface COM(1) Baud= 4800 Refresh= 60 minute
Nmb of event: 0
GPS Status : Error - GPS modul unreachable, no connect.
--- PCOS time services
TimeSyncDelay : OFF
Status time sync. : INVALID
Last time sync from UNKNOWN
--- Systeminformation
Operating in the daylight saving mode.
Time Zone : W. Europe Standard Time
Standard/Daylight: Bias -60/-60 minute
UTC Time : 04.09.2008 05:48
Local Time : 04.09.2008 07:48:59:0
\>

PCOS I/O services –SVCn

Wurde die Systemzeit über eine Service-Routine (IEC 60870-5-101 oder 104 Uhrzeitsynchronisation) gestellt, wird dies in dieser Rubrik angezeigt.

SNTP <NTP0:> services

Wurde die Systemzeit über einen Zeit-Server gestellt, wir dies hier angezeigt.

Die Funktion muss in der ALU520.INI frei geschaltet werden.

"SNTP service is not available" bedeutet, dass der Service in der ALU520.INI nicht freigegeben wurde,

GPS services

Zeigt den Status der Synchronisation über den Uhrzeitempfänger GPSRVC an.

Die Funktion muss in der ALU520.INI frei geschaltet werden. Siehe Kapitel 6.1.3.

"GPS service is not available" bedeutet, dass der Service in der ALU520.INI nicht freigegeben wurde,

8.4 ALU-Statusinformationen über Telnet

Über eine Telnet-Verbindung können Sie diverse Diagnosefunktionen aufrufen.

- Schließen Sie hierfür den Bedien-PC an die Ethernet-Schnittstelle auf der ALU an.
- Geben Sie im Windows Startmenü im Dialog Ausführen *telnet 192.168.0.52* ein (falls Sie die ALU-IP-Adresse geändert haben, geben Sie die geänderte Adresse ein).

In Dialogfenster das anschließend erscheint wird die WindowsCE Version und die Firmware-Version angezeigt.

☞ **Hinweis:** Die im Folgenden dargestellten Anzeigen sind nur Beispiele und können je nach ALU-Konfiguration von den tatsächlichen Anzeigen abweichen.

services list

Über diesen Aufruf können Sie feststellen, welche Treiber und Routinen auf der ALU gestartet sind.

Beispiel:

```
HTP0:   0x00030100      HTTPD.DLL       Aus
TEL0:   0x00031270      TELNETD.Dll     Wird ausgeführt
FTP0:   0x00031f10      FTPD.Dll        Wird ausgeführt
AUC0:   0x000320d0      Storage Card\A520_SYS\AX20UDPCFG.dll    Wird ausgeführt
B4D0:   0x00030500      Storage Card\PCOS_SYS\SSDKOSBIKA.dll    Wird ausgeführt
TEL1:   0x000386c0      telnetd.dll     Wird ausgeführt
```

diag_cpu

Zeigt ihnen Betriebsystem-, Hardware- und Treiber-Versionen an.

```
=========================================
OS Version          : 4.20 Build 0 Pb-Id 3
ALU Version         : 3.25 Build 0.1  04.02.10
PLC SYSTEM          : A250
PLC ALU             : ALU 520
ProConOS Version    : 4.40
Pcos Version        : 3.16
Primary class name  : CPU
Name of the board   : MODB
Manufacturer name   : OHP GmbH / JUMP
Date                : 5.2.2008
Serial number       : YXDKR0620
Hardware revsion    : 1.3
Firmware revsion    : 2.0
Jida interface rev.: 1.2
Jida boot count     : 31
-----------------------------------------
CPU Temperature     : 73.0°C (Sensor is operating)
CPU Voltage (core) : 0.961V (Sensor is operating)
DC  Voltage         : 3.489V (Sensor is operating)
DC  Voltage Standby: 3.489V (Sensor is operating)
```

Außerdem stehen Windows-Standardfunktionen wie z.B. ipconfig zur Verfügung.

ipconfig

```
Windows IP configuration
Ethernet adapter [PCI\RTL81391]:
         IP Address ........ : 192.168.0.51
         Subnet Mask ....... : 255.255.255.0

Ethernet adapter [PCI\E100CE1]:
         IP Address ........ : 192.168.0.70
         Subnet Mask ....... : 255.255.255.0
```

TEIL III Baugruppenbeschreibungen

1 Baugruppenbeschreibung ALU 504/518/520

1.1 Schnittstellen

Bezeichnung	Steckverbinder	Platzierung
1 Ethernet, **Programmierschnittstelle** *)	RJ 45	Frontseite, 1 von oben
1 Ethernet,	RJ45	Frontseite, 2. von oben
Serielle Schnittstelle COM 2 (RS232)	RJ 45	Frontseite, 3. von oben
Serielle Schnittstelle COM 1 (RS232)	RJ 45	Frontseite, 4. von oben
Diverse Anschlüsse siehe Kap. 1.2	Schraubklemmen	Frontseite
CF-Steckplatz	3M low Profile	Frontseite
2 USB Schnittstelle	2 USB Stecker	Frontseite
1 DVI Schnittstelle (ALU504 optional)	DVI-Stecker	Unterseite

*) **Hinweis:** Bei Auslieferung ist die Ethernet-Schnittstelle auf IP-Adresse 192.168.0.52 voreingestellt.

1.1.1 Schnittstellenbelegung der Schraubklemmen:

Klemme	Funktion	Beschreibung
1	UB	24V Betriebsspannung
2	UB	24V Betriebsspannung
3	Reserviert	
4	CAN_HZ	CAN High Zul. *)
5	CAN_LZ	CAN Low Zul. *)
6	CAN_GND	CAN Ground *)
7	CAN_GND	CAN Ground *)
8	Reserviert	
9	Reserviert	
10	GND	Ground Betriebsspannung
11	GND	Ground Betriebsspannung

*) in Vorbereitung

1.1.1.1 Stromversorgung

Die ALU 504/518 besitzt ein internes Netzteil mit einem Eingangsspannungsbereich von 14-30 VDC. Hierüber werden die ALU und die externen Peripheriekarten versorgt. Für die Peripheriebaugruppen, die über den PAB1 angeschlossen werden, stehen 8,5A zur Verfügung.

Im Unterspannungsfall < 14 V wird der Signalspeicher und der Telegrammpuffer der CPU auf der CF Card null-spannungssicher abgelegt.

1.1.1.2 Schnittstelle für GPS Uhrzeitempfänger

Die ALU504/518 besitzt eine Anschlussmöglichkeit für den optionalen GPS Uhrzeitreceiver GPSRCV von OHP. Dieser wird über eine der USB-Schnittstelle an die ALU504/518 angeschlossen.

1.1.1.3 CAN Schnittstelle (in Vorbereitung)

Die ALU 504/518 besitzt eine CAN Master/Slave – Schnittstelle. Die Anschaltung basiert auf dem Philips Chip SAJ1000. Die Schnittstelle ist mit Optokopplern galvanisch getrennt. Die Busadresse und die Terminierung werden über den 8-pol. DIL Schalter auf der oberen Frontseite der ALU eingestellt.

(siehe ALU 504/518 Bedienelemente)

1.1.2 Serielle RS232 Schnittstellen:

Pin-Nr.	Kurz Bez.	Signalbezeichnung	DIN-Bez.	DIN-Signalbezeichnung
1	DCD	Data Carrier Detect	M5	Empfangssignalpegel
2	RXD	Receive Data	D2	Empfangsdaten
3	TXD	Transmit Data	D1	Sendedaten
4	DTR	Data Terminal Ready	S1.2	DEE betriebsbereit
5	GND	Ground	E2	Betriebserde
6	+5VDC	Versorgung für GPS-Empfänger		
7	RTS	Request to Send	S2	Sendeteil anschalten
8	CTS	Clear to Send	M2	Sendebereitschaft

1.1.3 Compact Flash Speicherkarten

Die ALU 520 verfügt über einen Speicherkartensteckplatz, für eine CF-Karte. Die Karte wird unterhalb der LEDs eingesteckt. Es können Karten mit max. 8 GB Speichergröße verwendet werden.

Die CF-Karte enthält alle relevanten Daten für die ALU 520, wie z.B. Betriebs- und Laufzeitsystem, Anwenderapplikation und Telegrammpuffer. Die ALU wird mit einer betriebsfertigen CF- Karte ausgeliefert.

Achtung: Die CF-Karte nur im spannungslosen Zustand ziehen oder stecken. Formatieren Sie die von OHP ausgelieferten CF-Karten nicht, da hierdurch das Betreibsystem Windows CE gelöscht wird. Kopieren bzw. überschreiben Sie auch niemals die Datei CEBOOT.SYS im Stammverzeichnis der CF-Karte.

1.1.4 ALU504/518 Anzeigeelemente

Die ALU504/518 enthält 11 LEDs, die den Betriebszustand der ALU anzeigen

Nr.	Funktion	Farbe	Anschlussart
1	UB OK	grün	Dauerlicht
2	AWP Run, Stopp, Störung	grün	Dauerlicht, Dunkel, Blinkt
3	CAN OK, Fehler, NC	gelb	
4	Reserve	gelb	
5	Status Gold-Cap	rot	Dauerlicht = Kondensator nicht vollständig geladen
6	BAT1	rot	Dauerlicht = Batterie fehlt oder leer
7	BAT2	rot	Dauerlicht = Batterie fehlt oder leer
8	Reserve Erw. Karte	gelb	
9	Reserve Erw. Karte	gelb	
10	Reserve Erw. Karte	gelb	
11	Reserve Erw. Karte	gelb	

Während der Hochlaufphase nach Spannungs-EIN zeigen die LEDs folgendes Erscheinungsbild:

- LED 1…7 Dauerlicht
- Bei angeschlossenem Ethernet nach ca. 2 Sekunden grüne LED am RJ45 auf Dauerlicht, gelbe LED am RJ45 blinkt.
- Nach 40 - 120 Sekunden, je nach Ausführung der ALU, gehen die LED 3 bis LED 7 aus. Ist ein Anwenderprogramm als Bootprojekt auf der CF Card gespeichert wird dieses gestartet. Ist kein Anwenderprogramm geladen, geht die LED 2 ebenfalls aus.

1.1.5 ALU504/518 Bedienelemente

Reset Taster

Die CPU Platine enthält einen Reset-Taster auf der Vorderseite. Er befindet sich oberhalb des DIL Schalters und hat unabhängig von der CPU (ALU504/518) folgende Funktion:

- drücken des Tasters > 5s: Hardware Reset

- drücken des Tasters < 5s: Auslösung eines freien Interrupts

DIL Schalter

Der 8 pol. DIL-Schalter dient zur Einstellung des Startverhaltens sowie der CAN-Busadresse und der CAN Bus Terminierung (CAN Bus in Vorbereitung).

Der Schalter ist auf dem oberen Teil der Frontseite der ALU 504/518 montiert und kann übersichtlich bedient werden.

Schalter	Funktion
1	ON = CAN Bus abgeschlossen
2	ON = CAN Bus abgeschlossen
3	**ON = Warmstart, OFF = Kaltstart**
4	Reserviert
5	CAN Adr. (in Vorbereitung)
6	CAN Adr. (in Vorbereitung)
7	CAN Adr. (in Vorbereitung)
8	CAN Adr. (in Vorbereitung)

1.2 Technische Daten ALU504/518

Versorgung	
Versorgungsspannung	14V – 30V
Verlustleistung ALU504	< 15 W, ohne Erweiterungskarte
Verfügbarer Strom für E/A Teilnehmer	max. 8500mA
Serielle Schnittstellen	
Anzahl, Art	2, seriell, asynchron, V.24 Pegel,
Geschwindigkeit	max. 19.200 Baud
Anschluss	RJ45, Anschluss frontseitig
Statusanzeige	LED TXD/RXD im RJ 45 Stecker
Ethernet	
Anzahl, Art	2, IEEE802.3, 100 Base T, 100 MBit/s
Anschlussstecker	RJ45, Anschluss von Geräteseitefront
Statusanzeige	LED Link/Aktiv im RJ 45 Stecker
USB	
Anzahl, Art	2, Standard USB 2.0 frontseitig
PC Card Steckplatz	
Anzahl, Art	2, PC Card Typ 1 und 2 (PC Card nicht im Lieferumfang)
Monitoranschluss	
Anzahl, Art	1, DVI Anschluss (nur nutzbar mit neuem Magazin, bei ALU504 optional)
Sound	
Anzahl, Art	1, Line in, Line out, (nur nutzbar mit neuem Magazin)
CF Card Steckplatz	
Anzahl, Art	1, Standard CF Card, max. 8 Gbyte
CAN Anschluss	
Anzahl, Art	1 CAN Master,
Anschluss	Anschluss über Schraubklemmen frontseitig
CPU / Speicher	
Prozessortyp	ALU504 Intel Celeron 400 MHz
Speicher	ALU518 Intel Pentium M 1,1 GHz
Telegrammpuffer	128 MByte RAM / 256 MByte Ram
	1 MB im Flash, auf ALU keine Batterie erforderlich
Uhrzeitbehandlung	
Art	Gepufferte ms genaue RTC
Synchronisation	Über optionales GPS Modul über serielle Schnittstelle

CPU Erweiterungssteckplatz	
Anzahl, Art	1 Steckplatz für Erweiterungskarten
Schutzart	
Schutzart (IEC 60529) Schutzklasse (IEC61140) Isolation	IP20 III EN60950, IEC 950
Umweltbedingungen	
Umgebungstemperatur bei Betrieb Lagertemperatur Rel. Feuchte (IEC 68-2-1-1/2)	0°C bis 60°C -25°C bis 85°C bis 95% keine Betauung
Batteriepufferung	
Anzahl, Art Hinweis:	2 Typ AA 3,6V Pufferbatterien zur Pufferung der bestehenden PAB Teilnehmern, frontseitig getrennt wechselbar, Zustandsanzeige über LED, Anschluss für externe Batterie vorhanden. Die ALU benötigt selbst keine Pufferbatterien, da Pufferung relevanter Daten im Flash.
Bedienelemente	
Anzahl, Art	1 x 8pol. DIL Schalter für CAN Bus Anschluss und Adresse 1 x Reset Taster frontseitig bedienbar
Anzeigen	
11 LED-Anzeigen	1 LED UB OK 2 LED run, Stop, Störung 3 LED CAN Kommunikation OK/Fehler 4 LED res. 5 LED res. 6 LED Bat1 low 7 LED Bat2 low 8 LED res. Erw. Karte 9 LED res. Erw. Karte 10 LED res. Erw. Karte 11 LED res. Erw. Karte
Mechanischer Aufbau	
Baugruppe Format	Doppel-Europaformat 6 HE, 8 T

2 Baugruppenbeschreibung ALU 518-1/SFB und ALU 504-1/SFB

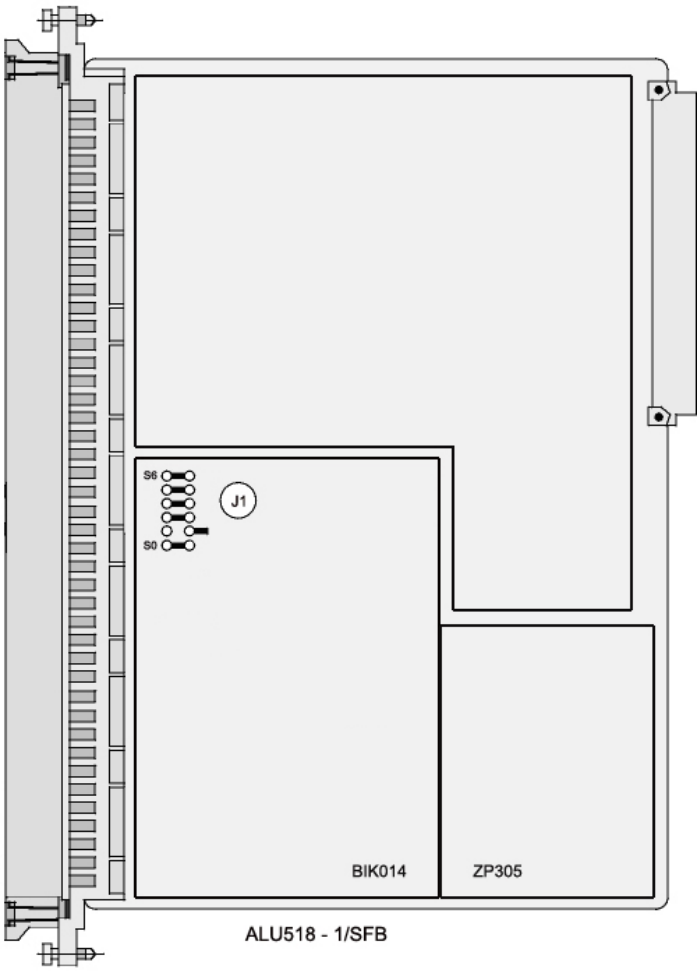
2.1 Allgemeines

ALU5xx-1SFB besteht aus der ALU5xx und der Zusatzplatine BIK 014 die zum Senden und Empfangen von Datentelegrammen über den Modnet 1/SFB dient. Sie stellt den Datentransfer zwischen Zentraleinheit und dezentralen E/A Einheiten her.

Technische Daten siehe Teil III, Kapitel 1.

☞ **Hinweis:** Die Firmwareversion vorhandener DEA 106 im dezentralen Baugruppenträger sollte den Änderungsindex 20 haben (kann vom Beschriftungsschild des EPROMS auf der DEA abgelesen werden). Bei älteren Firmwareständen kann es zu Problemen in der E/A-Kopplung kommen. Wenn nach der Umrüstung Ihre Anlage dieses Problem auftritt, wenden Sie sich an die OHP-Kundenbetreuung um ein Update für die Firmware zu erhalten.
EPROMS mit der Beschriftung "DEA105 Software Bitbus 042.217453.08" müssen auf jeden Fall ausgetauscht werden.

An der frontseitig zugänglichen RS485-Schnittstelle (untere Frontklemme von ALU) kann eine Bus-Linie mit bis zu 15 Teilnehmer (dezentrale Ein/Ausgaben – DEA-Prozedur – ALU5xx-1/SFB Master) angeschlossen werden.



ALU518 - 1/SFB

2.2 Bedienung / Darstellung

Die Frontseite der ALU wird um 2 Anzeigen ergänzt:

- 1 x grüne LED (L10) für Baugruppenfunktion

  leuchtet: keine Störung der Ausgangsspannung (U = 5 V) und Watch-dog der ALU hat nicht angesprochen die Baugruppe ist betriebsbereit (ready)

  erloschen: Störung der Ausgangsspannung (U = 5 V) oder Watch-dog der ALU hat angesprochen

- 1 x gelbe LED (L11) „comm“

  blinkt/ leuchtet: Wenn die BIK ein Telegramm (Daten- oder Statustelegramm) auf der Modnet 1/SFB-Schnittstelle sendet, leuchtet die LED auf (blinkt). Bei viel Telegrammverkehr ist durch die hohe Blinkfrequenz nur noch ein Dauerleuchten zu sehen.

  erloschen: Keine Telegramme auf der Modnet 1/SFB

2.3 Projektierung

Für ALU518-1SFB und BIK014 ist zu projektieren:

- Einstellen der Übertragungsrate
- Einstellung Master- oder Slave-Betrieb
- Adressierung bei Multiprog in der I/O-Konfiguration oder über ein Auswahlmenü im SPS-Konfigurator (nur bei Multiprog Vollversion mit OHP-Add-Ons).

Hinweis: OHP empfiehlt jedoch den Bestückungseditor des SPS-Konfigurators für die Projektierung in Multiprog zu verwenden, da hier die Ermittlung des korrekten Codes sehr komplex ist.

Projektierung mit dem SPS-Konfigurator

Um eine Bitbus-Schnittstelle in Multiprog zu definieren, muss der Bestückungseditor des SPS-Konfigurators geöffnet werden und ein neues Modul (BIK – Steckplatz A01) innerhalb der ALU518-Struktur (Steckplatz 001) erstellt werden. Mit dem BIK Modul können mehrere dezentrale Baugruppenträger verknüpft werden.

Die BIK Baudrate in Multiprog muss entsprechend der über die Steckbrücken eingestellten Baudrate projektiert werden

2.3.1 Koppel-Betrieb

Die Baudrate wird auf der interne BIK014 - Platine über Steckbrücken S0 und S1 eingestellt und muss innerhalb einer Feldbus-Linie einheitlich sein. Bei Geräten mit mehreren System-Feldbus-Kopplern kann jede Feldbus-Linie auf eine andere Baudrate eingestellt sein.

S0 S1	Bedeutung
	62,5 kBit/s; Asynchroner Betrieb
	375 kBit/s; Asynchroner Betrieb
	2 Mbit/s; Synchroner Betrieb
	Unzulässig

Die Bitbus-Betriebsart wird über die Steckbrücken S2 und S3 eingestellt.

S2	Bedeutung
	Master
	Slave

S3	Bedeutung
	Asynchroner Betrieb; Taktsignal nicht auf System-Feldbus geschaltet (2-Draht-Betrieb)
	Synchroner Betrieb; Taktsignal auf System-Feldbus geschaltet (4-Draht-Betrieb)

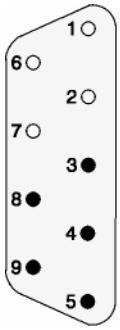
2.3.2 System-Feldbus-Anschluss

Das Buskabel zur Ankopplung der DEA an die ALU5xx – 1SFB muss vom Anwender selbst anfertigt werden. Das Kabel wird beginnend von der ALU durch Doppelbelegung der Anschlüsse von DEA zu DEA geführt (Buskopplung).

Anschluss an BIK014 Platine (D-Sub 9polig an ALU-Unterseite):

Steckerbelegung:

Kabel	**Pin**	**Signal bei**		
		62.5 kBd	**375 kBd**	**2 MBd**
Aderpaar 1	3	DATAN	DATAN	DATAN
	8	DATA	DATA	DATA
Aderpaar 2	4	-	-	DCLKN / RTSN
	9	-	-	DCLK / RTS
GND	5			



Pin	**Signal**	**Bedetung**
3	DATAN	Datensignale invertiert
4	DCLKN / RTSN	Taktsignal / Sendeaufforderung invertiert
5	RGND	Schutzerde, Abschirmung (R-Ground)
8	DATA	Datensignal
9	DCLK / RTS	Taktsignal / Sendeaufforderung

● Steckerpunkt belegt
○ Steckerpunkt nicht belegt

Anschluss an ALU-Frontseite (Schraubklemme):

Kabel	**Pin**	**Signal bei**		
		62.5 kBd	**375 kBd**	**2 MBd**
Aderpaar 1	42	DATAN	DATAN	DATAN
	41	DATA	DATA	DATA
Aderpaar 2	40	-	-	DCLKN / RTSN
	39	-	-	DCLK / RTS
GND	43			

Abschlusswiderstände

Sie können die Abschlusswiderstande mit dem DIL-Schalter auf der Unterseite der ALU bei Bedarf zuschalten.

Schalter	Zustand	Bedeutung
S1, S2, S3, S4	On	Widerstand 120 Ohm zugeschaltet
S1, S2, S3, S4	Off	Ohne Widerstand

Die Abschlusswiderstände werden jeweils am Anfang und Ende einer Verbindung benötigt.

Bei Einsatz vorhandener Bitbus-Kabel prüfen Sie, ob die Widerstände eventuell bereits in den BBS1-Steckern bestückt sind. In diesem Fall müssen die Brücken S1-S4 auf Off eingestellt werden.